Sistema de Predicción de Apuestas Deportivas: una aproximación a La Quiniela.

Grado en Ingeniería Informática

Autor: Francisco Javier Pérez Sánchez

Director: Araceli Sanchís de Miguel

Tutor: Germán Gutierrez Sánchez

Índice

Índice de figuras

Índice de tablas

1. Introducción

Las apuestas deportivas son un negocio que está en auge y que ha crecido mucho en los últimos años pese a la crisis. Dentro del mundo de las apuestas deportivas, existen dos formas de negocio distintas, las casas de apuestas y la Quiniela, aunque en esencia ambas líneas de negocio se basan en lo mismo, su diferencia primordial se basa en la forma de apostar.

Las casas de apuestas aparecieron en España en el 2005 y desde entonces es un negocio que ha ido creciendo progresivamente, sobre todo durante el periodo de 2009 a 2012 donde, a pesar de la crisis económica, la recaudación de las casas creció un 187%, moviendo cerca de los 400 millones de Euros en 2012. Las apuestas deportivas ofrecen una gran variedad de eventos deportivos sobre los que apostar (fútbol, carreras de galgos, etc.), ofreciendo la posibilidad al usuario de apostar directamente en un evento deportivo, combinaciones de varios o incluso sobre sucesos que puedan ocurrir durante el evento deportivo, como qué jugador marcará el primer gol, qué equipo irá ganando al descanso etc.

Los premios en las casas de apuestas se obtienen en función de la cantidad de dinero apostado y del coeficiente de beneficio en el momento de realizar la apuesta; así por ejemplo para el partido Real Sociedad - Real Madrid, la casa de apuestas Bet365 ofrece 1,75 € de beneficio por cada euro apostado si gana la Real Sociedad, por lo que si apostamos 10€ y acertamos obtendremos 17,5€. En definitiva las casas de apuestas son un mundo terriblemente amplio y complejo que cada año mueve más dinero.

La Quiniela es la gran competidora de las casas de apuestas deportivas en España, principalmente por dos motivos. El primero es el factor histórico y social de La Quiniela, la cual se inició en España en 1946 debido a la gran afición del pueblo español por el deporte rey, el fútbol, y siendo prácticamente la única entidad que en España ofrecía la posibilidad de apostar a eventos deportivos. Además La Quiniela destina el 45% de lo recaudado a beneficencia, por lo que efectúa una importante labor social. El segundo motivo es la modalidad de apuesta, La Quiniela a diferencia de las casas de apuestas, ofrece un número fijo y constante de apuestas semanales, ofreciendo un premio en proporción de lo recaudado, el número de aciertos y el número de acertantes. Sin embargo a diferencia de la tendencia de las casas de apuestas, La Quiniela ve reducida su recaudación año tras año desde que tocó techo en el año 2008 con más de 560 millones de €.

Con esto en mente no es de extrañar que la posibilidad de obtener cantidades ingentes de dinero atraiga a más de uno, si además podemos apostar de una forma relativamente segura (el factor azar siempre estará presente), y maximizar nuestros beneficios, mejor que mejor. Con estos propósitos en mente, un pequeño grupo de alumnos del grado en Ingeniería Informática, se propuso desarrollar una aplicación capaz de predecir el desenlace de un evento deportivo.

La idea de realizar una aplicación capaz de predecir resultados en eventos deportivos se originó durante el desarrollo de una asignatura del grado en ingeniería informática por los entonces alumnos Alejandro Vegas García, Andrés León Suarez Cetrulo y el autor de este TFG[18]. Como resultado de aquella cooperación se obtuvo una red neuronal capaz de predecir el resultado de los partidos de la Liga BBVA y la Liga Adelante, con una tasa de acierto superior al 50%.

1.1.Objetivos

El propósito principal de este trabajo fin de grado es el de generar una herramienta auto-contenida, capaz de predecir los desenlaces de los enfrentamientos de la primera y segunda división profesional de la liga española de fútbol, La Liga BBVA y La Liga Adelante; de esta forma podremos utilizar dichas predicciones para apostar en las distintas casas de apuestas o en La Quiniela.

Como punto de partida contamos con el trabajo previamente realizado por parte del autor y de sus compañeros Alejandro y Andrés[18], en la asignatura inteligencia artificial en las organizaciones, del grado en ingeniería informática de la universidad Carlos III de Madrid, por lo que basándonos en las conclusiones del trabajo anterior intentaremos obtener mejores resultados en las predicciones, además se creará una base de conocimiento propia para almacenar la información que creamos relevante y poder sacar el máximo rendimiento posible de ella gracias a la minería de datos.

Para poder alcanzar los objetivos indicados, se proponen los siguientes objetivos concretos:

a) Creación base de conocimiento propia.
b) Desarrollo sistema de predicción.
c) Herramienta auto-contenida.

Creación base de conocimiento propia

El principal motivo de crear una base de conocimiento propia, es porque la recopilación y almacenamiento de información relacionada con el problema es uno de los pilares en la minería de datos. Ya que el principal objetivo de la minería de datos es el de analizar de forma automática o semi-automática ingentes cantidades de datos y detectar posibles patrones, como podrían ser dependencias no lineales entre los datos, datos atípicos, etc.

Por este motivo crearemos una base datos en el lenguaje MySQL, porque este lenguaje ha demostrado a lo largo de los años su robustez e integridad a la hora de crear y gestionar bases de datos, además es software libre por lo que no tendremos que preocuparnos de licencias y dispone de una amplia comunidad de usuarios que además ofrecen amplio soporte ante posibles incidencias que podamos sufrir.

Es importante remarcar la relevancia de este objetivo, ya que el éxito o fracaso del predictor que desarrollemos estará estrechamente ligado a nuestra base de datos, porque de ella se extraerán los datos que nos permitirán encontrar un modelo capaz de predecir los resultados de un partido de fútbol, así que cuanto más completa y detallada sea nuestra base de datos, mejores probabilidades tendremos de encontrar un modelo adecuado.

Desarrollo de un sistema de predicción

Este objetivo consiste en encontrar un sistema de predicción, basado en técnicas propias de la Inteligencia Artificial (IA) capaz de obtener el resultado final de un partido perteneciente a cualquiera de las primeras dos divisiones del fútbol español, conocidas como La Liga BBVA y la Liga Adelante.

Para realizar este objetivo se recurrirá a técnicas de minería de datos, las cuales nos ayudarán a generar un modelo de predicción basado en los datos utilizados. También utilizaremos la herramienta de software libre conocida como WEKA (Waikato Environment for Knowledge Analysis [13]), la cual dispone de gran cantidad de algoritmos y filtros pre-configurados para la tarea.

Herramienta auto-contenida

Este objetivo consiste en desarrollar una aplicación capaz de ejecutarse en cualquier sistema que reúna unos requisitos mínimos; de esta forma conseguimos que la herramienta sea portable y podamos ejecutarla desde cualquier dispositivo, con las implicaciones comerciales que eso conlleva.

1.2.Motivación

Los motivos principales de este trabajo fin de grado son los siguientes:

a) Implementar, aplicando técnicas y algoritmos propios de dos áreas distintas: búsqueda y recuperación de la información, y del aprendizaje automático, una herramienta con utilidad comercial, adquiriendo experiencia en el desarrollo de aplicaciones de cara al mercado laboral.
b) Realizar un trabajo interesante con probabilidad de obtener un rendimiento económico de él.
c) Involucración por parte del alumno en un proyecto en el que convergen intereses y hobbies del alumno.

1.3.Planificación

A continuación se muestra la planificación del proyecto, mostrando el desglose de las tareas que lo han compuesto, el orden en el que se han realizado, el tiempo dedicado y los recursos utilizados.

En las siguientes 2 figuras (Figura 1 y Figura 2) se muestra la lista final de tareas del proyecto:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	**Análisis Inicial**	**6 días**	**vie 03/01/14**	**vie 10/01/14**		**Ingeniero**
2	Objetvos TFG	3 días	vie 03/01/14	mar 07/01/14		
3	Definición del Sistema	3 días	mié 08/01/14	vie 10/01/14	2	
4	**Diseño Inicial**	**18 días?**	**lun 13/01/14**	**mié 05/02/14**	**1**	**Ingeniero**
5	Análisis tecnologías	5 días	lun 13/01/14	vie 17/01/14		
6	Análisis de requisitos	5 días?	lun 20/01/14	vie 24/01/14	5	
7	Diseño Estructura de la Aplicación	5 días?	lun 27/01/14	vie 31/01/14	6	
8	Diseño de la base de datos	3 días?	lun 03/02/14	mié 05/02/14	7	
9	**Implementación**	**22 días?**	**jue 06/02/14**	**vie 07/03/14**	**4**	**Progamador**
10	Implementación Crawler	6 días?	jue 06/02/14	jue 13/02/14		
11	Implementación Parseador HTML	6 días?	vie 14/02/14	vie 21/02/14	10	
12	Almacenamiento Info BDFutbol en BBDD	6 días?	lun 24/02/14	lun 03/03/14	11	
13	Implementación Parseador Partidos	1 día?	vie 21/02/14	vie 21/02/14		
14	Almacenamiento Partidos en BBDD	1 día?	lun 24/02/14	lun 24/02/14	13	
15	Comprobación Integridad BBDD y	4 días?	mar 04/03/14	vie 07/03/14	14;12	

Figura 1: Lista de tareas del proyecto, parte 1

15	Comprobación Integridad BBDD y	4 días?	mar 04/03/14	vie 07/03/14	14;12	
16	**Algoritmo Rendimiento Jugadores**	**4 días**	**mié 12/03/14**	**lun 17/03/14**	**9**	**Ingeniero**
17	Análisis Valoración Jugador	2 días	mié 12/03/14	jue 13/03/14		
18	Implementación Valoración Jugadores	1 día	vie 14/03/14	vie 14/03/14	17	Progamador
19	Actualización de la Base de conocimiento	1 día	lun 17/03/14	lun 17/03/14	18	Progamador
20	**Implementación Técnica Data-Mining**	**16 días?**	**mar 18/03/14**	**mar 08/04/14**	**16**	**Ingeniero**
21	Estudio técnicas Data Minning	1 día?	mar 18/03/14	mar 18/03/14		
22	Análisis Diseño RNA	1 día?	mié 19/03/14	mié 19/03/14	21	Progamador
23	Implementación RNA	7 días	jue 20/03/14	vie 28/03/14	22	Progamador
24	Evaluación	7 días	lun 31/03/14	mar 08/04/14	23	Progamador
25	**Integración**	**9 días**	**mié 09/04/14**	**lun 21/04/14**	**20**	**Ingeniero**
26	Diseño aplicación	2 días	mié 09/04/14	jue 10/04/14		
27	Implementación	7 días	vie 11/04/14	lun 21/04/14	26	Progamador

Figura 2: Lista de tareas del proyecto, parte 2

En las siguientes 3 figuras (Figura 3, Figura 4 y Figura 5) se muestra el diagrama de Gantt correspondiente a la lista de tareas:

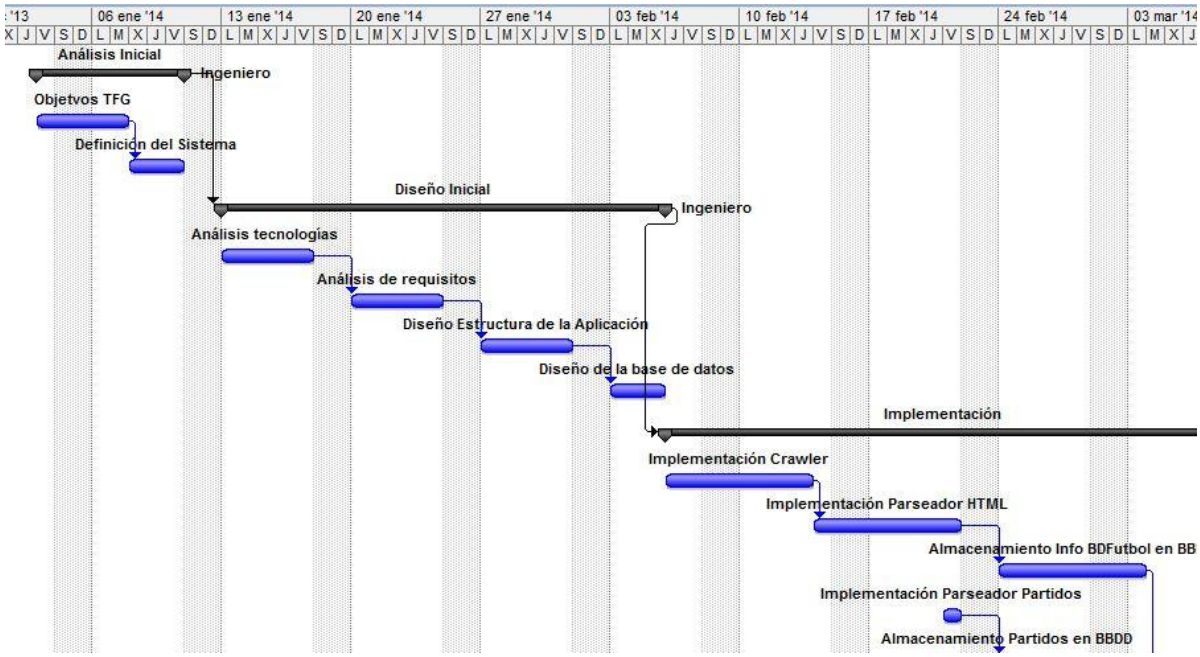

Figura 3: Diagrama de Gantt del proyecto, parte 1

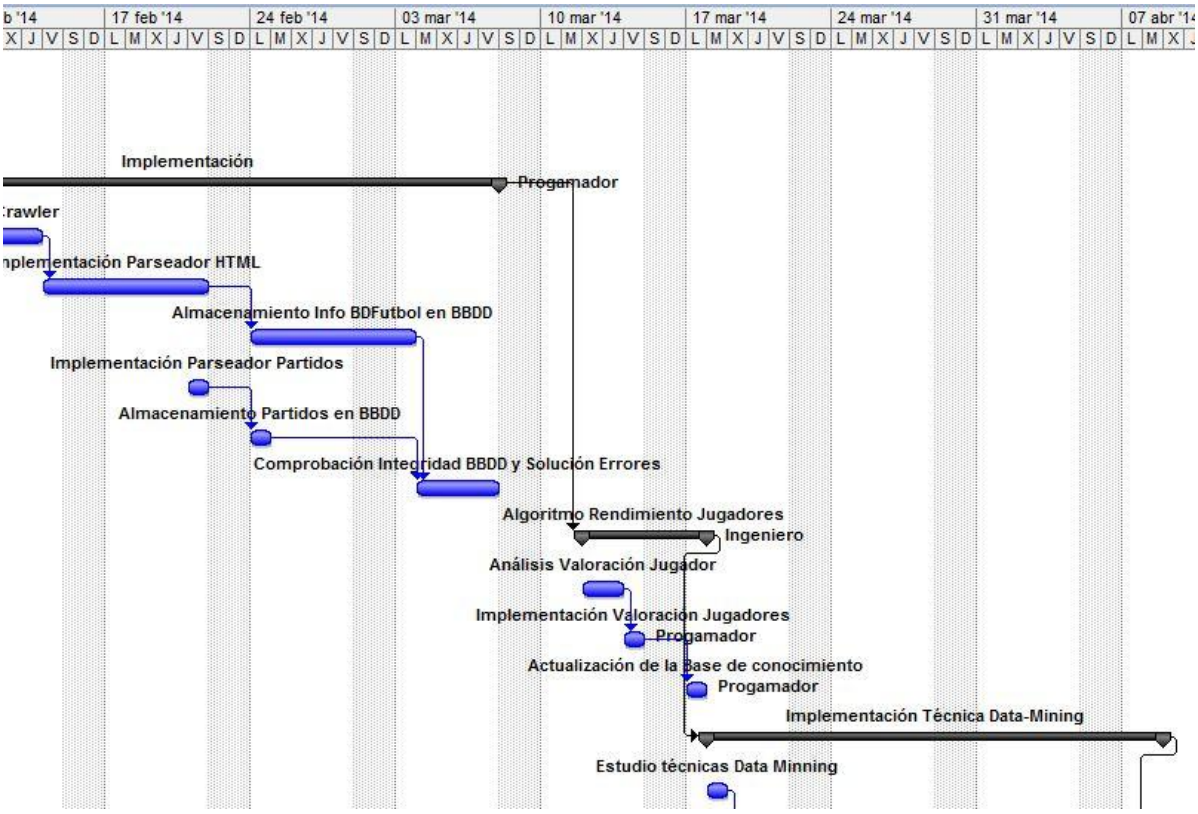

Figura 4: Diagrama de Gantt del proyecto, parte 2

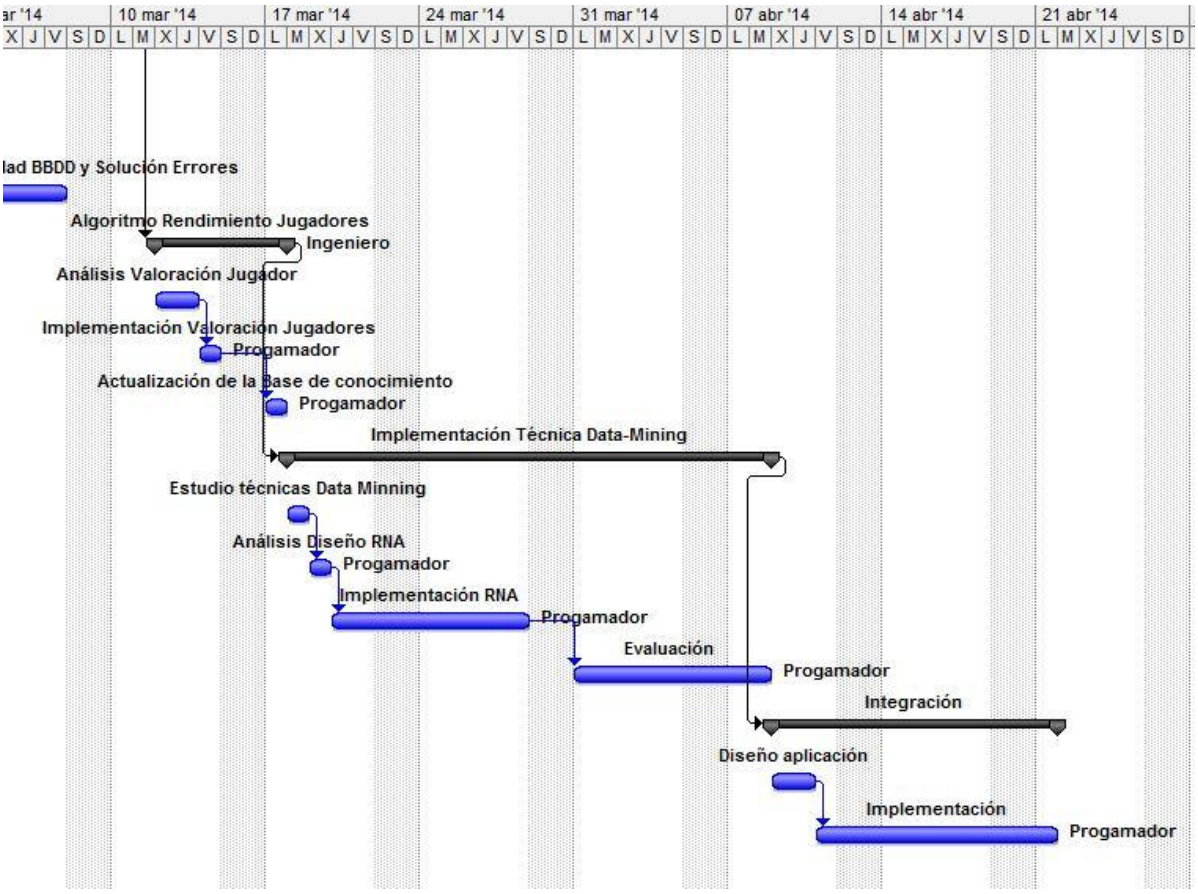

Figura 5: Diagrama de Gantt del proyecto, parte 3

1.4.Presupuesto

En este apartado se detallan los costes del proyecto, tanto costes de personal, como en material o fungibles.

Costes de personal:

Dentro de los costes de personal se engloban los gastos en forma de salarios que se abonarán a los miembros del proyecto.

La siguiente figura contiene el desglose de las horas dedicadas a cada una de las tareas del proyecto por los distintos miembros que lo componen:

Id	Recurso	Horas
1	Ingeniero	200 horas
	Análisis Inicial	48 horas
	Análisis tecnologías	40 horas
	Análisis requisitos	40 horas
	Diseño estructura aplicación	24 horas
	Diseño base de datos	24 horas
	Algoritmo rendimiento jugador	16 horas
	Diseño estrategia data mining	8 horas
2	Programador	336 horas
	Implementación crawler	48 horas
	Implementación parseador HTML	48 horas
	Supervisión almacenamiento datos	8 horas
	Implementación parseador partidos	8 horas
	Supervisión almacenado partidos	8 horas
	Integridad BBDD y corrección errores	40 horas
	Implementación valoración jugadores	8 horas
	Actualización BBDD y comprobación	8 horas
	Implementación RNA	56 horas
	Evaluación	56 horas
	Implementación herramienta	48 horas

Tabla 1: Horas tarea y recurso

Para el cálculo de los salarios se ha establecido un precio hora en bruto de 15€ para el programador, y de 25€/hora para el ingeniero.

Recurso	Coste/hora	Trabajo	Coste
Ingeniero	25 €	200 horas	5000 €
Programador	15 €	336 horas	5040 €
Total		**536 horas**	**10040€**

Tabla 2: Presupuesto personal

Tal y como podemos observar los costes presupuestados asociados al personal del proyecto ascienden a 10040€.

Costes materiales:

Los costes materiales engloban desde la amortización del hardware informático necesario para el proyecto, hasta los materiales fungibles y demás gastos derivados del proyecto.

Para el cálculo de la amortización tanto del software como del hardware utilizado aplicaremos la siguiente ecuación:

$$\frac{A}{B} \times C$$

A : Tiempo de uso, en meses, del elemento para el proyecto.

B : Periodo de depreciación del elemento, en meses. En nuestro caso lo estableceremos en 60 meses para elementos de Hardware y 36 para elementos Software.

C : Precio sin IVA del elemento.

Descripción	Coste (€)	Tiempo dedicado	Periodo depreciación	Coste Proyecto(€)
Ordenador Portátil	578 (x2)	3,35 (x2)	60	64,54
Weka	0	3,35	36	0
MySql 5	0	3,35	36	0
Crawler 4J	0	3,35	36	0
Jsoup	0	3,35	36	0
Windows 7 Profesional	106,61	3,35	36	9,92
Ubuntu 12.04	0	3,35	36	0
MS Office 2010	400	3,35	36	37,22
MS Project	0	3,35	36	0
Total				**111,68**

Tabla 3: Coste amortizado asociado

El coste total derivado tanto de hardware como del software del proyecto asciende a 111,68€.

Descripción	Coste (€)
Fungibles	80
Amortización	11,68
Total	**191,68**

Tabla 4: Presupuesto material

Resumen:

Una vez calculados los costes de personal y material del proyecto, solo falta calcular el coste total del proyecto, que además de los costes anteriores incluye un pequeño porcentaje referido a los costes indirectos que se pueden derivar del proyecto, tales como facturas de luz, alquiler de oficina y demás imprevistos que puedan surgir durante la realización del proyecto. Para cubrir dichos costes indirectos, hemos incrementado el coste del mismo en un 10%.

Descripción	Total
Personal	10040
Material	191,68
SubTotal	**10231,68**
Costes Indirectos	1023,17
Total	**11254,85**

Tabla 5: Presupuesto

Como podemos observar el coste total del proyecto asciende a 8271,31 €.

1.5.Medios empleados

Los medios empleados para el correcto desarrollo del trabajo son los siguientes:

- Hardware:
  - Estación de trabajo: Intel Core i7 2670QM, 8GB de RAM (X2)
  - Conexión ADSL 1Mb
- Software y Librerías utilizadas:
  - Sistemas Operativos:
    - Ubuntu 12.04
    - Windows 7 Professional
  - Lenguajes de Programación:
    - Java 1.7
    - MySQL 5.0
  - Entornos de desarrollo integrado (IDE):
    - MySQL Worbench
    - Netbeans 7.4
  - Librerías Java:
    - Jsoup 1.7.3
    - Crawler 4j
    - Weka 3-7-11
  - Otros
    - Virtualbox 4.3.10

2. Estado del arte

La aplicación de técnicas de inteligencia artificial en asuntos de la vida cotidiana es cada vez más común, desde el uso de algoritmos de *pathfinding* para robots de limpieza a algoritmos de aprendizaje en videojuegos.

El dominio de este proyecto se centra en el análisis de grandes volúmenes de información. Esta información se genera a diario y que puede ser de los temas más variopintos, meteorología, economía, etc. En ocasiones, si se analizan esos volúmenes de datos, podemos encontrar asociaciones y patrones que no son visibles a simple, por lo que es necesario procesar esos datos, modelarlos y aplicar diversas técnicas de análisis para poder encontrar esos patrones e identificarlos.

El análisis manual de la información es caro, porque es un procedimiento lento y tedioso que requiere una gran cantidad de recursos humanos, los cuales pueden formulan hipótesis, probarlas, ajustarlas, reformular dichas hipótesis o generar nuevas y así sucesivamente. Es en este punto donde entran en juego las distintas técnicas de Inteligencia artificial, más concretamente aquellas técnicas orientadas a la minería de datos.

La minería de datos es una de las áreas de investigación que se engloban de la ingeniería de la computación. Este área consiste en desarrollar modelos o técnicas capaces de encontrar relaciones entre cantidades ingentes de datos, que a priori pueden no revelar ningún tipo de relación, de esta forma se pueden predecir comportamientos basados en hechos anteriores y detectar patrones.

Este capítulo tiene por finalidad poner al lector en antecedentes, para ello se realiza una breve puesta en escena de la situación actual en las distintas áreas de la sociedad con las que este proyecto está relacionado.

2.1.Inteligencia Artificial (I.A.)

El término inteligencia artificial fue acuñado por primera vez en 1956, durante un congreso en el cual se hicieron previsiones muy optimistas a corto plazo sobre investigaciones recientes en el campo de la computación. La imposibilidad de alcanzar aquellas previsiones o ni siquiera de aproximarse a ellas, junto con las limitaciones técnicas de la época, provocaron que este campo fuese un terreno estuviese estigmatizado para los investigadores; no es hasta mediados de los 60 cuando empiezan a aparecer resultados de investigaciones con aplicaciones reales para la época, las cuales parecieron animar a determinados grupos de investigadores, que poco a poco han profundizado en la materia, desarrollando técnicas cada vez más avanzadas y complejas.

El concepto de IA es aún hoy día demasiado difuso, algunos investigadores se refieren a la IA como a la ciencia encargada de imitar el comportamiento del cerebro humano, otros se refieren a máquinas inteligentes sin sentimientos que puedan obstaculizarles para hallar la solución a un problema, etc. El punto que tienen en común casi todas las teorías es que la IA debe ser capaz de ayudar al ser humano a resolver problemas de cualquier tipo.

Es quizá por ese motivo que la mayoría de investigadores se centran principalmente en la faceta orientada a la resolución de problemas, la lógica o la razón, pero existe cada vez una

tendencia más numerosa que aboga por incorporar componentes emotivos con el fin de aumentar la eficacia de los sistemas inteligentes, así mismo es cada vez más numerosa la literatura de ciencia ficción que aborda el tema, llegando incluso a tratarse estos temas en la gran y pequeña pantalla[1].

El motivo por el que consideremos al género de ciencia ficción relevante en este punto, es porque gracias a los escritores de ciencia ficción (quién no ha escuchado hablar nunca de Isaac Asimov) y a su imaginación los investigadores han profundizado en temas de I.A, en ocasiones buscando hacer posible lo que sólo antes existía en un los libros, realizando de esta forma avances en algunos campos, como la robótica, término que se originó en la literatura (Isaac Asimov).

En la actualidad gracias a los avances tecnológicos, que han aumentado la potencia de cómputo de los ordenadores, el incremento del volumen de información generada diariamente y al avance de las técnicas de I.A. es posible desarrollar sistemas auxiliares de apoyo a prácticamente cualquier tarea cotidiana, y es por eso que creemos que la I.A jugará un papel relevante en el futuro más inmediato de la sociedad.

Este proyecto se basa en los estudios realizados hasta la fecha en la área de la I.A. dedicada al aprendizaje automático. Dentro del campo del aprendizaje automático existen multitud de técnicas desarrolladas a lo de los años, las cuales se pueden clasificar en dos grandes grupos, en técnicas de aprendizaje supervisado y en técnicas de aprendizaje no supervisado. Las técnicas de aprendizaje no supervisado difieren de las técnicas de aprendizaje supervisado por el motivo de que no es necesario disponer de conocimiento previo del problema.

2.2.Aplicaciones conocidas

Es difícil encontrar en el mercado aplicaciones capaces de predecir con un elevado grado de fiabilidad resultados deportivos, aunque estamos seguros de que existen, porque existen artículos que versan sobre el tema y son multitud los proyectos o publicaciones que tratan de abordar la situación con mayor o menor éxito.

Existe la posibilidad de que la dificultad de encontrar una descripción detallada de tales sistemas venga derivada del posible impacto económico que estas aplicaciones podrían . El dinero que se mueve en el mundo de las apuestas es muy alto (más de 1000 millones de € en el 2013, en España) y por lo tanto entendemos que el hermetismo por parte de las empresas que desarrollan este tipo de sistemas sea elevado. A pesar de ello, hemos sido capaces de encontrar artículos periodísticos que hablan con poco nivel de detalle sobre estos sistemas. De la bibliografía existente hemos identificado un sistema., *Víctor the predictor* **¡Error! No se encuentra el origen de la referencia.**

Víctor the predictor

Este sistema conocido como *Víctor the predictor* pertenece a la casa de apuestas Betfair[2].Lo poco que se deduce de los artículos que en los que se cita o se describe, es que es un sistema basado en una Red de Neuronas Artificiales que utiliza esta casa de apuestas para predecir resultados de partidos de fútbol. Como muestra de la eficacia de este sistema, en el 2008 predijo que la Selección española de fútbol ganaría la Eurocopa 2008, circunstancia que pocos expertos pudieron ni siquiera imaginar a principio del campeonato.

[1] Almost Human

Aplicaciones móviles

En el mercado de las APP (aplicaciones) móviles existen multitud de aplicaciones que permiten realizar apuestas deportivas y que además ofrecen datos históricos, coeficientes de probabilidad (basados en datos ofrecidos por las casa de apuestas) o incluso predicciones, pero tras revisar comentarios de usuarios que supuestamente han utilizado estas aplicaciones, llegamos a la conclusiones que los sistemas "predictores" no son más que estrategias de marketing, herramientas aleatorias o tal vez en los mejores casos utilicen realmente modelos matemáticos para predecir dichos resultados.

Entre todas las aplicaciones móviles encontradas que ayudan a realizar La Quiniela hemos decidido analizar la mejor valorada por los usuarios, Quinidroid. Esta aplicación dispone de una gran funcionalidad relacionada con el mundo de las quinielas, porque permite realizar una quiniela desde la propia plataforma a través de la aplicación web de la página oficial de La Quiniela, hacer un seguimiento en tiempo real de la evolución de la jornada, además ofrece datos estadísticos sobre los resultados seleccionados por otros usuarios de la aplicación, histórico de enfrentamientos, datos de webs externas que ofrecen pronósticos de la quiniela y una predicción técnica que no sabemos en qué está basada.

2.3.Teoría de apuestas

Antes de profundizar más en la temática de la teoría de las apuestas, es conveniente que nos familiaricemos con la terminología utilizada en este área:

- El ***bankroll*** es el dinero disponible para apostar.
- El ***stake*** es porcentaje o cantidad de nuestro *bankroll* destinado en la apuesta.
- La **probabilidad estimada** se refiere a una valoración, subjetiva en la mayoría de los casos, sobre la probabilidad de ganar.
- La **Cuota** es el rendimiento por euro apostado que ofrece la casa.

A continuación describimos las tres técnicas principales en el ámbito de las apuestas: el Criterio de Kelly, Martingale, Sistema Apuesta Proporcionada

Criterio de Kelly

Una de las teorías más utilizadas por los apostadores es el criterio de Kelly[16]. Esta teoría recibe este nombre por su fundador John L. Kelly, un matemático aficionado a las apuestas de caballos, que en 1956 escribió un artículo llamado *"A new interpretation of the Information Rate"*, basado en el trabajo de Claude Shannon.

El objetivo de esta estrategia es el de maximizar el crecimiento del *bankroll*, determinando el porcentaje de *bankroll* que debemos destinar a cada apuesta. Para calcular este porcentaje debemos aplicar la siguiente fórmula:

Stake =(Cuota x (Probabilidad estimada/100)-1) / (Cuota -1) x 100

[2] Betfair

En función del porcentaje de *stake* obtenido apostaremos, por lo que cuando obtengamos un valor negativo o nulo, nos abstendremos de apostar, también debemos tener presente que el sistema no es perfecto, y que presenta algunas carencias críticas que deberías tener en cuenta, como son:

1. La probabilidad estimada, este parámetro al ser subjetivo, condiciona enormemente la eficiencia de este método, por lo que el éxito o fracaso de este método depende altamente de nuestro criterio a la hora de asignar un valor a este parámetro.

2. El *stake* puede resultar a veces muy elevado, por lo que se recomienda el uso de coeficientes que lo reduzcan.

Martingale

Martingale[17] es un sistema de apuestas anterior al criterio de Kelly desarrollado por en los casinos y es bastante simple. El sistema consiste en incrementar la cantidad apostada cada vez que se pierde, para de ésta forma compensar las pérdidas y sacar beneficio tras ganar la primera apuesta, tras la cual se vuelve a apostar la cantidad inicial.

La ecuación que indica la cantidad a apostar cada vez que se pierde es la siguiente:

$$stake = (Pérdidas+Beneficio) / (Couta -1)$$

Siendo *Beneficio,* el beneficio que se desea obtener y *Pérdidas* el acumulado de dinero perdido hasta la fecha. El mayor problema que plantea este sistema es que no se suele disponer de un bankroll ilimitado para cubrir las apuestas cuando el ratio de pérdidas + beneficio es elevado.

Apuesta proporcionada

Este sistema de apuestas consiste en variar la cantidad apostada en función del tamaño del *bankroll*, para calcular la apuesta el usuario aplica un porcentaje fijo de su *bankroll* a cada apuesta, de forma que si el *bankroll* es grande, la apuesta será alta y si el *bankroll* es pequeño la apuesta será baja. La ecuación que determina el *stake* apostado será la siguiente:

$$stake = Bankroll \times Porcentaje\ aplicado$$

El parámetro clave de este sistema es el porcentaje del *bankroll* aplicado, los expertos en la materia recomiendan establecer este valor entre el 1% y el 5% del *bankroll*.

2.4.Minería de datos

Como ya sabemos, las ciencias de la computación abarcan multitud de campos, siendo uno de esos campos la minería de datos. Este campo de la computación consiste en la aplicación de técnicas de aprendizaje automático, estadística y análisis de bases de datos sobre conjuntos de información para extraer patrones ocultos a simple vista.

Este área es relativamente joven porque siempre ha estado limitada por el hardware y por las cantidades de información disponible, es ahora en la actualidad cuando este área está revelando su auténtico potencial, porque cada vez aparecen máquinas más potentes, algoritmos de procesado más eficientes y tecnologías capaces de procesar volúmenes de información cada vez mayores (Big Data).

La temática de los datos a los que esta área de la computación puede ser aplicada es muy amplia, desde predicciones meteorológicas a la búsqueda de zonas óptimas para la instalación de plantas petrolíferas.

Probablemente la mayor limitación a la que se enfrenta la minería de datos viene definida por el conjunto de datos utilizado para la resolución del problema. Esto es así porque si se utiliza un conjunto de datos corrupto, incompleto o que no sea representativo del problema que queremos solucionar, afectará gravemente y limitará el modelo generado, sin importar la técnica aplicada, por este motivo es necesario asegurarse de utilizar un conjunto de datos adecuado.

Afortunadamente se han desarrollado técnicas o métodos capaces, no de resolver los problemas de un conjunto de datos, pero sí de mitigar las posibles deficiencias más comunes que presente un conjunto de datos, por ejemplo si disponemos de conjuntos de datos que presenta registros incompletos se pueden utilizar modelos de estimación o predicción para rellenar los huecos de los registros, etc.

Los especialistas en la materia, por norma general, suelen cumplir una serie de procedimientos comunes a la hora de realizar un análisis:

- Selección y análisis del conjunto de datos a utilizar. Este punto suele ser crítico porque a partir de aquí cualquier resultado obtenido estará condicionado por el conjunto de datos original.
- Pre-procesado del conjunto de datos. En este apartado se suele preparar el conjunto de datos seleccionado para aplicar la técnica de data mining seleccionada.
- Construcción del modelo. En este punto ya se ha escogido que técnica o técnicas de aprendizaje automático se van a utilizar y se construye el modelo.
- Evaluación de datos. Este suele ser el paso final de la tarea; es en este momento cuando se analiza si el modelo generado cumple o no con las expectativas y si es necesario volver al punto de partida y aplicar otro enfoque distinto.

Algunas de las técnicas de computación que se suelen aplicar en esta materia son: Redes neuronales, *clustering* y árboles de decisión.

Por último indicar que los puntos que quizá están cobrando más relevancia en la actualidad y que posiblemente afecten a la evolución de este campo son la integración de estas técnicas en sitios web y el procesado en tiempo real. De hecho cada vez aparecen más herramientas orientadas al análisis de datos en *streamming*, como podría ser MOA (*Massive Online Analysis)* una herramienta que está siendo desarrollada en la actualidad por la universidad de Waikato.

2.5.Apuestas deportivas La Liga

Instaurada en 1946, La Quiniela ha monopolizado todo el mundo relacionado con las apuestas deportivas en España, no es hasta 2005 cuando empiezan a aparecer las primeras alternativas a La Quiniela. El primero de estos competidores en asentarse en el mercado Español fue la conocida casa de apuestas Betfair. La aparición de esta casa de apuestas fue toda una revolución en España por su sistema novedoso de apuestas, ya que permitía apostar a un evento deportivo de forma individual, mientras que La Quiniela obligaba a los apostantes a predecir al menos 10 de los 15 resultados que ofrecía para poder acceder a los premios.

2.5.1. Loterías y Apuestas del estado

El primer sistema de apuestas legal conocido en España aparece en 1946. La aparición de este sistema marca un antes y un después en el mundo de los juegos de azar, porque cubría un nuevo nicho de mercado en la administración de loterías y apuestas del estado, que en aquella época era más conocida por la Lotería Nacional.

Este nuevo sistema adquirió popularidad rápidamente porque permitía apostar en el deporte rey español, el fútbol, y porque destinaba un 45% de la recaudación a beneficencia. Es importante hacer constar el hecho de que un alto porcentaje de la recaudación era destinada a fines sociales, porque hacía relativamente pocos años España había salido de una guerra y el nivel económico en la época era bastante precario.

En estos primeros años de vida de La Quiniela la dificultad de los pronósticos era muy elevada porque requería que el apostante, además de acertar el desenlace del encuentro debía también acertar el resultado del marcador. Este sistema duró solamente dos años antes de cambiar al conocido sistema del "1x2" que aún hoy se utiliza.

El primer sistema de premios de La Quiniela era bastante complejo. Este sistema consistía en asignar un número de puntos basándose en la similitud o exactitud del resultado predicho con el realmente acontecido. La puntuación asignada era la siguiente: 30 puntos en el caso de que el resultado fuera exacto, 20 si se acertaba el ganador y la misma diferencia de goles, 19 si había diferencia de un gol, 18 puntos si había diferencia de 2 goles. En caso de empate, un gol de diferencia sobre el resultado real suponía 19 puntos y dos goles de diferencia 18 puntos.

Dos años más tarde, en 1948, este sistema cambia radicalmente, simplificándose el sistema de apuesta y el de puntuación; pasando a ser 14 los partidos a predecir, 8 de la primera división y 6 de la segunda categoría, siendo necesario predecir solamente el desenlace del partido; para ello se adoptó el sistema "1X2" el cual codifica los posibles desenlaces de un partido de la siguiente forma: el 1 para indicar la victoria local, la x indicaba un empate y el 2 victoria del equipo visitante, siendo esta codificación la utilizada hoy día.

La cantidad de partidos a predecir ha variado ligeramente a lo largo de los años, de los 14 partidos instaurados en 1948, se pasó a 15 en 1988 y ese es el modelo que se sigue utilizando actualmente. Este decimoquinto partido solamente se tiene en cuenta si se aciertan los otros 14; sino se aciertan los otros 14 partidos, este partido no computa como acierto.

La siguiente tabla (tabla6) se muestra el porcentaje de la recaudación destinado a los premios:

Categoría	Aciertos	Porcentaje Recaudación
Pleno al 15	15	10%
1ª	14	12%
2ª	13	8%
3ª	12	8%
4ª	11	8%
5ª	10	9%

Tabla 6: Distribución premios La Quiniela

Como podemos observar en la tabla 6, el número mínimo de aciertos necesario para poder acceder a los premios es 10.

Tipos de apuesta

La Quiniela actualmente permite varias modalidades de apuestas:

- **Apuesta simple**: Este tipo de apuesta se basa en diversas combinaciones de hasta catorce dobles, se puede incluir dos pronósticos por casilla, y 9 triples, tres pronósticos por casilla, todo incluyendo de forma única por boleto el pleno al quince, que no podrá ser compuesto por dobles o triples. El riesgo y cantidad monetaria aumenta, pero del mismo modo aumenta la probabilidad de ganancia
- **Apuesta múltiple Reducida**: El apostante elige las diferentes combinaciones de dobles y triples, las posibilidades aumentan y el precio disminuye, pero por el contrario, las combinaciones de dobles y triples se deben al azar, contrarrestando el efecto posibilidad. Hay seis tipos diferentes de apuestas múltiples reducidas.
- **Apuesta múltiple condicionada:** se trata del último tipo de apuesta, tiene una mayor dificultad de entendimiento, y consiste en jugar únicamente con algunas de las apuestas que se considera con mayor probabilidad de acierto. Se marca el número de dobles y triples, teniendo en cuenta varios partidos en conjunto. Es necesario marcar al menos dos partidos a condicionar de entre los seleccionados con dobles y triples. Las condiciones tendrán en cuenta el número total de aciertos en cuanto a número de variantes (X, 2), número de empates o número de doses, acertados en total, de entre los partidos seleccionados como participantes en la condición. La regla que marca la excepción, es la imposibilidad de que el pleno al quince participe como partido condicionado.

Para terminar esta breve introducción sobre la Quiniela, creemos que es necesario analizar la evolución de la recaudación de La Quiniela en los últimos años, concretamente queremos mostrar el efecto económico que las casas de apuestas parecen estar teniendo en las recaudaciones anuales de La Quiniela. En la Figura 6 mostramos la evolución en la recaudación anual de La Quiniela.

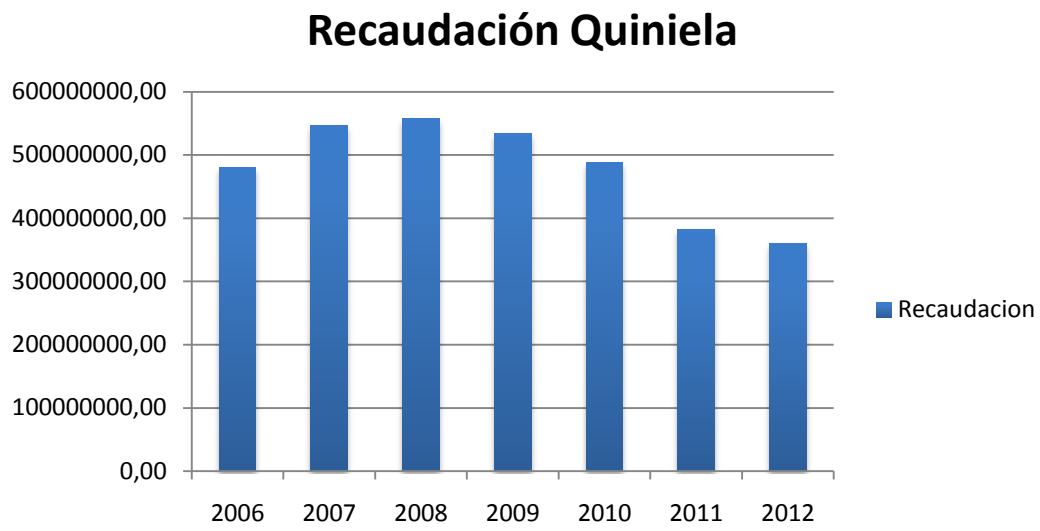

Figura 6: Gráfico recaudación La Quiniela

Tal y como podemos observar en la figura 6, la tendencia en la recaudación de La Quiniela tiende a reducirse año a año desde el 2008, esta tendencia puede estar motivada por el ámbito de crisis económica de los últimos años, pero probablemente también se deba a la irrupción en el mercado español de las casas de apuestas, rival directo en el sector de la apuesta deportiva de La Quiniela.

2.5.2. Casas de apuestas

El sistema de apostado de las casas de apuestas difiere sobremanera con respecto a la Quiniela, mientras que el sistema anterior es un sistema rígido en el que el apostante siempre apuesta a los mismos partidos y el reparto de beneficios se determina en función de la recaudación, el sistema de las casas de apuestas es mucho más ágil y atractivo para el apostante, ya que permite realizar apuestas de eventos más específicos y el reparto de beneficios depende de la cantidad apostada en función de la cuota que ofrece la casa de apuestas en el momento de la apuesta.

La cuota de una apuesta representa el coeficiente de beneficio que se obtendrá por cada euro apostado, así un coeficiente de 1,2 nos devolverá 1,2 euros por cada euro apostado, ganando 0,2€ por euro.

Si tuviésemos que remontarnos a la aparición de este sistema de apuestas, viajaríamos al periodo clásico en el cual, según los historiadores, los griegos de la Antigua Grecia ya apostaban en las distintas disciplinas de sus eventos deportivos, más conocidos como Juegos Olímpicos. A mediad que avanzamos en la historia, los historiadores han podido encontrar rastros evidentes de que en distintos lugares a lo largo de todo el mundo, las distintas culturas existentes tenían sistemas de apuestas, algunos de ellos legales y por tanto regularizados por las leyes de la época y en otro no tanto.

Pero no es hasta finales del siglo XVIII cuando este tipo de sistemas empieza a cobrar una gran popularidad gracias a las entorno socio-cultural de la época. Gracias al colonialismo y la

inmigración un mismo modelo de apuesta se expande ampliamente, popularizándose y provocando que en algunos periódicos de la época surjan secciones dedicadas a las apuestas deportivas.

A pesar de que el modelo de apuesta es muy popular, no es hasta principios del siglo XX, cuando aparecen los primeros locales de apuestas deportivas, expandiéndose rápidamente. Pero aún así el público al que se podía acceder era relativamente limitado, y no es hasta la explotación de Internet cuando las casas de apuestas han revelado todo su potencial.

En España, concretamente, el modelo de apuesta ofrecido por las casas de apuestas, era marginal hasta hace poco tiempo, porque existía La Quiniela, un modelo de apuesta más popular que ostentaba su particular monopolio, y que crecía año a año según se incrementaba el poder adquisitivo del español medio.

Pero en el año 2005 y gracias a internet, aparece en el mercado español la primera casa de apuestas importante de este tipo, Betfair, compitiendo duramente desde un principio con La Quiniela, ganando poco a poco terreno al monopolio hasta conseguir desbancar en la actualidad a La Quiniela como el sistema de apuestas que más dinero recauda.

3. Sistema Propuesto

3.1.Sistema propuesto

Para la creación de la herramienta ha sido necesario el uso de distintas tecnologías que ha habido que integrar.

En primer lugar ha sido imprescindible la creación de una base de conocimiento lo suficientemente saturada e íntegra como para poder aplicar técnicas de Data mining, para ello se ha requerido la extracción de información de fuentes externas fiables [1]

A continuación se ha decidido almacenar dicha información en una base de datos relacional en MySQL, que es una tecnología Open Source fiable, robusta, altamente contrastada y accesible para este tipo de tareas.

Por último se ha desarrollado una herramienta en Java capaz de extraer la información necesaria en cada momento de la base de datos y procesarla de forma eficaz para obtener las predicciones deseadas.

La figura 7 muestra un diagrama de flujo de la aplicación:

Figura 7: Diagrama Flujo Aplicación

La aplicación a desarrollar se puede fragmentar en dos secciones que pueden trabajar de forma independiente si se desea. Por un lado estaría la creación de la base de conocimiento y manipulación de los datos para el sistema de predicción y por otro lado estaría la parte encargada de desarrollar un sistema predictivo fiable a partir de los datos ofrecidos la sección encargada del tratamiento de la información.

3.2.Marco Regulador

Este proyecto al recoger y almacenar información de jugadores, entrenadores y entidades jurídicas reales debe acogerse a la Ley Orgánica de Protección de Datos (LOPD). Además se ha decidido que para una correcta eficiencia y eficacia en el desarrollo del proyecto es recomendable seguir determinadas metodologías y estándares, los cuales describimos a continuación.

3.2.1. Metodolgía

Para un correcto desarrollo del proyecto, se ha decidido usar la metodología desarrollada por Barry Boehm, o más conocida como metodología en espiral[4]. Boehm describió su metodología como un modelo de desarrollo cíclico, en el que en cada iteración aumenta la complejidad y el riesgo del modelo. Cada ciclo se divide en 4 fases o actividades, relacionadas con las distintas fases del diseño software, planificación, determinación de objetivos, análisis de riesgos y desarrollo.

La figura 8 muestra un diagrama de la metodología en espiral desarrollada por Barry Boehm

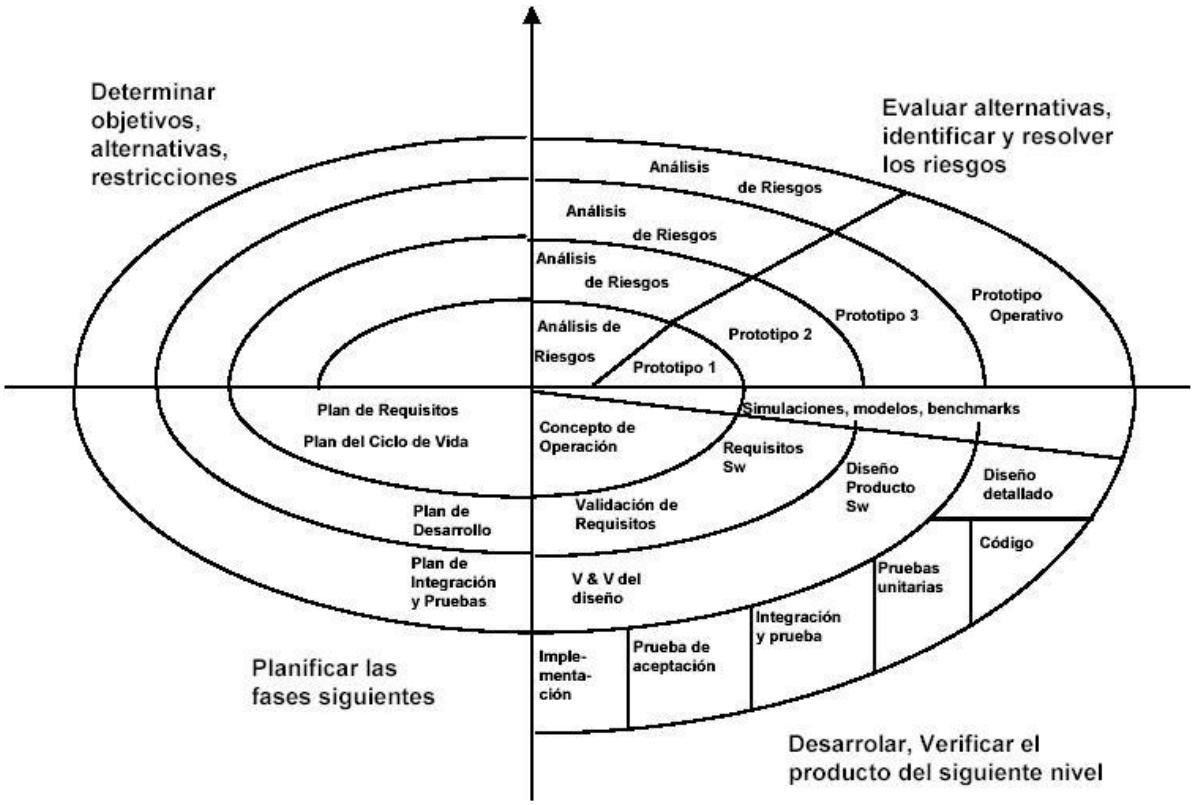

Figura 8: Figura representativa de metodología en espiral

3.2.2. Estándares

El uso de estándares es un punto clave para la organización y comprensión de las tareas de desarrollo, de esta forma se facilita enormemente la legibilidad del código a los desarrolladores, tanto los involucrados en el proyecto como terceros, ayudando a reducir costes en futuras modificaciones de código. Para la codificación de este proyecto se han propuesto los siguientes estándares.

Estándar de Java

Estos son los puntos más relevantes del estándar de codificación de Java:

- La extensión de los ficheros de código es '.java' y la extensión de las clases compiladas '.class'.

- Cada fichero sólo contiene una clase o interfaz pública. En caso de usar alguna clase o interfaz privada se pueden incluir en el mismo fichero, pero el fichero siempre debe
- contener al principio del mismo la clase o interfaz pública.
- La unidad de sangrado es 4 espacios y la tabulación debe tener 8 espacios.
- La longitud de las líneas de código no debe superar los 80 caracteres, ya que líneas de más de 80 caracteres no se muestran correctamente en muchas herramientas.
- Cuando una línea queda incompleta se deben seguir los siguientes principios:
  - Terminar la línea después de una coma.
  - Terminar la línea antes de un operador.
  - Es preferible tener un mayor sangrado en la siguiente línea.
  - La siguiente línea debe comenzar a la misma altura que el inicio de la expresión en la anterior línea.
- Los comentarios pueden tener el siguiente formato:
  - Comentario de línea: dos barras ('// ...').
  - Comentario de varias líneas: comienza con '/*' y termina con '*/', empezando cada línea con un asterisco:

    ```
    /*
     Comentario
     */
    ```

  - Comentario 'javadoc': sirve para crear la documentación de la API (*Application Programming Interface*) de nuestro programa. Comienza con '/**' y termina con '*/', empezando cada línea con asterisco al igual que el comentario de varias líneas.
- Es recomendable incluir comentarios javadoc antes de cada método y variable.
- Es recomendable tener una única declaración de variable por línea.
- Sólo debe haber una sentencia por línea.
- Las variables se deben declarar al comienzo del método.
- Las variables se deben inicializar en la declaración, excepto si su valor inicial depende del código siguiente.
- El espaciado de paréntesis, corchetes y llaves es el siguiente:
  - Los corchetes deben aparecer sin espacios.
  - Las llaves siempre deben tener un espacio.
  - Los paréntesis se escriben sin espacio si contienen los parámetros de un método y con espacio si forman parte de una sentencia (if, while, etc.).
- Los operadores deben espaciarse.
- Las llaves de apertura '{' deben estar en la misma línea que la sentencia y las de cierre '}' en la siguiente.
- Se deben introducir líneas en blanco entre métodos, entre las variables locales y el cuerpo del método y entre diferentes secciones lógicas de código.
- El nombrado de variables y clases sigue las siguientes reglas:
  - Las clases e interfaces se escriben con mayúscula inicial.
  - Los métodos y variables se escriben con minúscula inicial.
  - Las constantes se escriben completamente en mayúsculas.

- Para separar palabras dentro del nombre de una variable se puede usar el carácter '_' iniciar la siguiente palabra con mayúscula excepto para las constantes.

Estándar MySql

En este apartado se enumeran una serie de normas de estilo y buenas prácticas para la construcción de modelos de datos relacionales.

- Los nombres de las tablas y campos deben estar compuestos por caracteres alfanuméricos, evitando los espacios en blanco o tabulaciones y recomendando reservar el carácter barra baja '_' para los nombres compuestos o la alternancia de minúsculas y mayúsculas.
- Es aconsejable que los nombres usados, tanto para las tablas como para los campos no supere los 32 caracteres.
- El nombre de las tablas así como el de las columnas es aconsejable que estén escritos en singular.
- El nombre de las tablas además de estar escrito en singular, debe representar la información que se almacena en ella.
- El nombre de las columnas debe ser lo suficientemente descriptivo.
- Cuando el valor de la columna dependa de una tabla distinta (claves primarias, etc), el nombre de la columna debe iniciarse por el nombre de la tabla origen, seguido del carácter de separación que se esté utilizando y por último el nombre de la columna:

  nombreTablaOrigen_nombreColumna
- Todas los nombres de clave deben utilizar el prefijo "id".
- En las consultas empotradas en el código Java de la aplicación, se recomienda reflejar en letra mayúsculas las palabras clave del lenguaje MySQL, por ejemple:

  *"SELECT * FROM Liga.Equipo WHERE idEquipo = x"*

Estándar DOM

Document Object Models (DOM), es básicamente un estándar dedicado a representar objetos HTML o XML y que está regulado por el *World Wide Web Consortium* (W3C). El estándar DOM se originó inicialmente para que los programas pudiesen manipular el contenido de documentos HTML y XML.

Debido a la evolución de la web y los navegadores, el estándar DOM ha ido evolucionando para adaptarse a las necesidades de cada momento y evitar de esta forma problemas de compatibilidad, obligando al organismo regulador W3C, a emitir varias especificaciones sobre el estándar, las cuales hasta la fecha son, DOM nivel 0, DOM nivel 1, DOM nivel 2 y DOM nivel 3.

El uso del estándar DOM en nuestra aplicación se centra en las mejoras basadas en las especificaciones de la versión DOM nivel 2, lanzada en Noviembre del año 2000. Dichas mejoras se basan en la manipulación de eventos del navegador y la manipulación de partes del texto en las páginas. Estas mejoras nos permiten extraer el texto plano de los objetos DOM por medio de sus etiquetas identificadoras, así como identificar los enlaces integrados en las distintas páginas web que analizaremos.

Los puntos más relevantes del estándar de codificación DOM son:

- Las extensiones de los ficheros de código es ".html" para los documentos que no contengan ningún tipo de código PHP, la extensión ".php" se utilizará en los documentos que contengan cualquier tipo de codificación en el estándar PHP, la extensión ".js" es exclusiva de los documentos que contienen el código javascript y la extensión ".css" será utilizada por los ficheros que contienen las normas de estilo.
- Indicar en la cabecera del documento todas las librerías javascript, hojas de estilo y variables globales que se van a utilizar.
- Añadir las etiquetas de metadatos oportunas para la correcta identificación y clasificación del documento.
- Indicar el título del documento.
- Mantener un sangrado en el documento que permita identificar las dependencias jerárquicas.
- No exceder 80 caracteres por línea del documento.
- Utilizar identificadores descriptivos para los elementos del documento.

3.3.Requisitos software

Después de analizar el estado del arte vamos a analizar los requisitos software necesarios para llevar a cabo el proyecto. Es recomendable prestar atención a esta fase del proyecto porque un fallo en los requisitos puede provocar un fallo en cascada que si no se detecta a tiempo puede acarrear un incremento en los costes del proyecto.

Los requisitos los clasificaremos en dos categorías, los requisitos funcionales (RF) que son aquellos que engloban las distintas funcionalidades de la herramienta y los requisitos no funcionales (RNF) que están relacionados con las restricciones.

Utilizaremos la siguiente tabla para definir los requisitos:

Nombre:	RF/RNF-idRequisito
Resumen:	Breve descripción del requisito
Necesidad:	Obligación de cumplir el requisito, sus valores pueden ser: Alta/Media/Baja
Prioridad:	Relevancia del requisito, puede tomar los cualquiera de los siguientes valores: Alta/Media/Baja
Descripción:	Descripción detallada del requisito

Tabla 7: Tabla de requisitos

3.3.1. Requisitos funcionales

Nombre:	RF-001
Resumen:	Información BBDD
Necesidad:	Alta
Prioridad:	Alta
Descripción:	Se permite indicar el servidor, usuario y contraseña de la base de datos.

Tabla 8: Requisito RF-001

Nombre:	RF-002
Resumen:	Filtrado Atributos
Necesidad:	Alta
Prioridad:	Alta
Descripción:	Se podrá elegir si se desea aplicar filtrado de atributos en el modelo o no.

Tabla 9: Requisito RF-002

Nombre:	RF-003
Resumen:	Selección parámetros algoritmo
Necesidad:	Alta
Prioridad:	Alta
Descripción:	Se permite cambiar los parámetros de configuración del algoritmo seleccionado para generar el modelo.

Tabla 10: Requisito RF-003

Nombre:	RF-004
Resumen:	Carga jornada
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La aplicación recibirá los partidos de la jornada en un fichero con formato propio

Tabla 11: Requisito RF-004

Nombre:	RF-005
Resumen:	Salida Predicción
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La predicción de la herramienta se devolverá en un fichero de texto, en la ruta indicada, con el nombre "Predicciones.txt"

Tabla 12: Requisito RF-005

Nombre:	RF-006
Resumen:	Directorio de trabajo
Necesidad:	Alta
Prioridad:	Alta
Descripción:	A la herramienta se le debe indicar un directorio de trabajo, donde se generarán los modelos, ficheros de entrenamiento, test y las predicciones.

Tabla 13: Requisito RF-006

Nombre:	RF-007
Resumen:	Selección modelo
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La herramienta permite al usuario seleccionar el modelo que desee conveniente para las predicciones.

Tabla 14: Requisito RF-007

Nombre:	RF-008
Resumen:	Creación BBDD
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La herramienta debe poder crear la base de datos que necesita para generar los modelos que el usuario desee, siempre dentro de las capacidades que ofrece la herramienta.

Tabla 15: Requisito RF-008

3.3.2. Requisitos no funcionales

Nombre:	RNF-001
Resumen:	Lenguaje implementación herramienta
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La herramienta se desarrollará en el lenguaje de programación Java

Tabla 16: Requisito RNF-001

Nombre:	RNF-002
Resumen:	Base de datos
Necesidad:	Alta
Prioridad:	Alta
Descripción:	Se ha seleccionado una base de datos relacional basada en el lenguaje MySQL

Tabla 17: Requisito RNF-002

Nombre:	RNF-003
Resumen:	Partidos fichero entrada
Necesidad:	Alta
Prioridad:	Alta
Descripción:	Pese a que la quiniela tiene un número fijo de partidos, la herramienta podrá recibir un número variable de partidos como entrada, siempre y cuando se clasifiquen correctamente

Tabla 18: Requisito RNF-003

Nombre:	RNF-004
Resumen:	Formato fichero entrada
Necesidad:	Alta
Prioridad:	Alta
Descripción:	El fichero de entrada será un fichero de texto, con caracteres reservados para separar los partidos de primera división de los de segunda

Tabla 19: Requisito RNF-004

Nombre:	RNF-005
Resumen:	Predicciones por partido
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La herramienta solamente devolverá una predicción por partido.

Tabla 20: Requisito RNF-005

Nombre:	RNF-006
Resumen:	Fichero de Salida
Necesidad:	Alta
Prioridad:	Alta
Descripción:	La herramienta generará las predicciones en un fichero de salida con un nombre determinado por la propia herraienta.

Tabla 21: Requisito RNF-006

3.4.Búsqueda, extracción y almacenamiento de la base de conocimiento

Para llevar a cabo la creación de la base de conocimiento se han usado dos aplicaciones open source de Java para la extracción y una base de datos en MySQL para almacenar la información.

Así todo, el proceso de creación de la base de conocimiento o base de datos, se resume en:

- Búsqueda de fuentes de información fiables
- Extracción de dicha información
- Almacenamiento de la información

La figura 9 representa un diagrama de flujo de esta tarea:

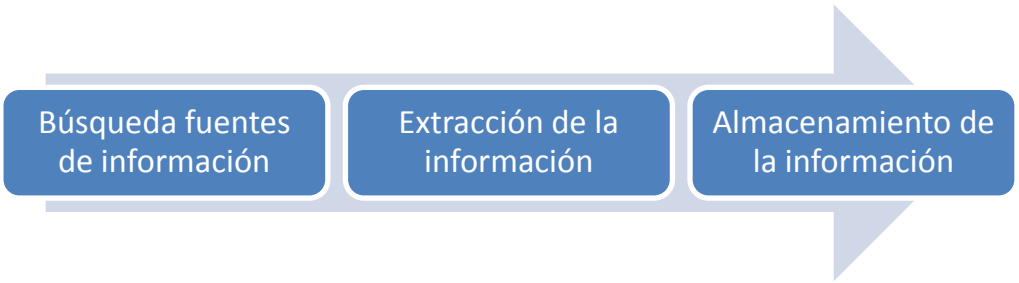

Figura 9: Diagrama flujo búsqueda, filtrado, extracción y almacenamiento de la información

3.4.1. Búsqueda

El primer paso para la creación de la base de datos, ha consistido en encontrar distintas fuentes de información en la web con información útil para nuestro propósito, que además tuviesen un fácil acceso, fuesen fiables y no tuviesen licencias restrictivas que nos impidiesen su explotación. En este punto encontramos distintas webs con información sobre las distintas divisiones de La Liga española de fútbol, pero muchas estaban altamente incompletas, eran de dudosa procedencia o no proporcionaban la posibilidad de extraer la información de forma automatizada.

En la siguiente tabla (tabla 22), se resumen las fuentes de información encontradas:

Dominio	Descripción	Utilizada
http://www.lfp.es/	Toda la información de la Liga BBVA y Liga Adelante, horarios, resultados, clasificación, noticias, estadísticas, y mucho más...	No, Solo ofrece información detallada de la temporada actual y para acceder al histórico se requiere que el crawler utilice javascript.
http://www.ligabbva.com/	LigaBBVA.com - Página oficial de la mejor liga de fútbol del mundo, la liga BBVA	No, Página dedicada casi exclusivamente a las noticias, apenas ofrece información estadística relevante.
http://www.bdfutbol.com	Base de Datos de Primera División (Liga BBVA), Segunda División (Liga Adelante), Segunda B y Selección Española. Temporadas, plantillas, jugadores, entrenadores	Sí, Ofrece información estadística histórica relevante, además de una fácil disponibilidad de extracción y existe información relacionada con entrenadores, árbitros etc.
http://www.rankinghistorico.es	Rankings y estadisticas historicas acumuladas de la Liga de futbol española. Ranking por afición, peñas, estadios, presupuestos, goles...	No, Numerosas relaciones estadísticas que podemos aplicar, pero pocos datos para añadir a la BBDD.
www.loteriasyapuestas.es	Página oficial de La Quiniela	Sí, Ofrece posibilidad de acceder al histórico, recaudaciones, etc.
http://data.betfair.com/	Histórico apuestas de Betfair	No, No cumplimos los requisitos para poder acceder a ellos.
http://liga.host56.com/	Histórico partidos 1ª y 2ª División Española.	Sí, Se mantiene actualizada y contiene todos los partidos de 1ª y 2ª división española desde 1972.
http://www.losmillones.com/	Histórico de quinielas con sus respectivos premios, además de otras métricas relacionadas con las apuestas y los equipos.	Sí, Fuente de conocimiento estadístico.

Tabla 22: Relación webs con información

Una vez encontradas las webs que podían resultar útiles, se analizó qué información era prioritaria; para este primer filtrado de información se tuvieron en cuenta distintos parámetros, como la exactitud de la información, la fiabilidad del sitio, la posibilidad de acceso y extracción automático y nuestra propio criterio para clasificar la información que 'a priori' pudiera ser relevante. Con todo eso decidimos quedarnos con dos páginas webs, una que contenía los partidos de fútbol de 1ª y 2ª división desde la temporada 1972-73, que además permitía descargarlos en un fichero de texto [11] y la otra web, que ofrecía de forma estructurada la gran mayoría de jugadores, entrenadores, árbitros y equipos de 1ª, 2ª y 2ªB [1] y que además parecía ser fiable, ya que en la página principal de dicha web se indicaba la colaboración de un periódico deportivo de tirada nacional.

Las figura 10, 11 y 12 representan la completitud de los datos de forma visual

Leyenda colores

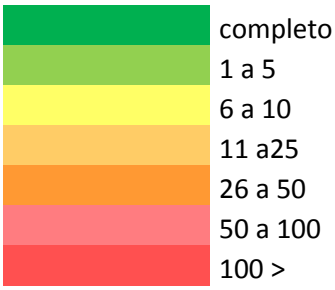

Figura 10: Leyenda

Primera División

Jugadores

Temporada	Jugadores	Sin Nombre Completo	Sin Datos	Sin Foto
1928-29	210	11	40	12
1929-30	221	9	34	17
1930-31	206	16	37	18
1931-32	202	12	33	12
1932-33	214	13	32	19
1933-34	216	13	33	15
1934-35	261	16	35	24
1935-36	266	12	34	23
1939-40	262	6	18	28
1940-41	270	10	11	20
1941-42	314	9	13	13
1942-43	323	3	4	8
1943-44	323	0	2	6
1944-45	322	0	1	4
1945-46	333	2	6	17
1946-47	301	1	3	14
1947-48	299	1	2	9
1948-49	288	0	1	10
1949-50	312	2	5	11
1950-51	352	0	3	6
1951-52	357	0	4	7
1952-53	355	0	3	0
1953-54	365	1	2	6
1954-55	372	0	3	8
1955-56	362	0	2	14
1956-57	354	1	1	9
1957-58	369	0	1	4
1958-59	370	0	1	0
1959-60	384	0	1	1
1960-61	365	0	0	0
1961-62	400	0	1	6
1962-63	401	0	2	4
1963-64	428	1	2	0
1964-65	397	0	0	0
1965-66	415	0	0	1
1966-67	395	0	1	0
1967-68	402	0	2	0
1968-69	390	0	0	0
1969-70	401	0	1	0
1970-71	404	0	1	1
1971-72	438	0	2	0
1972-73	433	0	0	0
1973-74	447	0	0	0
1974-75	444	0	0	0
1975-76	447	0	0	0
1976-77	446	0	1	0
1977-78	442	0	0	0
1978-79	453	1	0	1
1979-80	459	0	0	0
1980-81	449	1	1	0
1981-82	484	0	0	0
1982-83	458	1	0	2
1983-84	453	0	0	0
1984-85	621	8	9	39
1985-86	451	0	1	2
1986-87	465	0	0	5
1987-88	512	0	0	1
1988-89	523	0	0	0
1989-90	508	0	0	1
1990-91	522	0	0	0
1991-92	517	0	0	1
1992-93	501	0	0	0
1993-94	502	0	0	0
1994-95	519	0	0	0
1995-96	585	0	0	0
1996-97	624	0	0	0
1997-98	573	0	0	0
1998-99	581	0	0	0
1999-00	583	0	0	0
2000-01	596	0	0	0
2001-02	576	0	0	0
2002-03	591	0	0	0
2003-04	637	0	0	0
2004-05	640	0	0	0
2005-06	648	0	0	0
2006-07	624	0	1	0
2007-08	619	0	1	2
2008-09	621	0	0	0
2009-10	623	0	1	0
2010-11	635	0	2	1
2011-12	631	0	0	0
2012-13	647	0	0	0

Entrenadores

Temporada	Entrenadores	Sin Nombre Completo	Sin Datos	Sin Foto
1928-29	11	2	4	2
1929-30	10	1	2	1
1930-31	10	1	3	1
1931-32	10	1	2	1
1932-33	10	0	0	0
1933-34	10	0	1	0
1934-35	12	0	1	0
1935-36	16	0	3	1
1939-40	15	0	1	0
1940-41	13	0	0	2
1941-42	17	0	1	1
1942-43	17	0	1	0
1943-44	14	0	0	0
1944-45	16	0	0	0
1945-46	17	0	3	1
1946-47	15	0	2	0
1947-48	16	0	2	1
1948-49	15	0	1	2
1949-50	17	0	0	1
1950-51	20	0	1	1
1951-52	20	0	2	2
1952-53	23	0	3	2
1953-54	19	0	0	0
1954-55	21	0	1	0
1955-56	22	1	2	2
1956-57	18	0	0	0
1957-58	21	0	0	0
1958-59	27	0	0	0
1959-60	29	0	0	0
1960-61	31	0	0	0
1961-62	28	0	2	1
1962-63	24	0	0	0
1963-64	23	0	0	0
1964-65	28	1	1	1
1965-66	22	0	0	0
1966-67	26	0	2	0
1967-68	23	0	0	0
1968-69	24	0	0	0
1969-70	29	1	1	1
1970-71	22	0	1	0
1971-72	26	0	1	0
1972-73	21	0	0	0
1973-74	21	0	0	0
1974-75	24	0	1	0
1975-76	23	0	0	0
1976-77	22	0	0	0
1977-78	24	0	0	0
1978-79	24	0	0	0
1979-80	28	0	0	0
1980-81	26	0	0	1
1981-82	27	0	0	0
1982-83	26	0	0	0
1983-84	26	0	0	0
1984-85	24	0	0	0
1985-86	27	0	0	0
1986-87	30	0	0	0
1987-88	31	0	0	0
1988-89	42	0	0	0
1989-90	34	0	0	0
1990-91	37	0	0	0
1991-92	27	0	0	0
1992-93	41	0	0	0
1993-94	34	0	0	0
1994-95	37	0	0	0
1995-96	41	0	0	0
1996-97	48	0	0	0
1997-98	34	0	0	0
1998-99	36	0	0	0
1999-00	29	0	0	0
2000-01	30	0	0	0
2001-02	27	0	0	0
2002-03	32	0	0	0
2003-04	30	0	0	0
2004-05	30	0	0	0
2005-06	32	0	0	0
2006-07	27	0	0	0
2007-08	34	0	0	0
2008-09	30	0	0	0
2009-10	31	0	0	0
2010-11	29	0	0	0
2011-12	45	0	0	0
2012-13	31	0	0	0

Figura 11: Datos 1ª División

Segunda División

Jugadores

Temporada	Jugadores	Sin Nombre Completo	Sin Datos	Sin Foto
1959-60	717	109	207	189
1960-61	710	91	187	160
1961-62	699	65	157	118
1962-63	759	86	182	164
1963-64	728	71	162	126
1964-65	691	49	109	110
1965-66	725	54	95	104
1966-67	741	49	87	83
1967-68	773	33	57	57
1968-69	499	13	43	27
1969-70	491	10	34	30
1970-71	501	10	36	21
1971-72	501	6	18	15
1972-73	507	8	21	27
1973-74	511	12	32	18
1974-75	512	11	34	27
1975-76	510	7	26	15
1976-77	493	15	36	20
1977-78	485	13	16	24
1978-79	503	14	11	28
1979-80	550	20	28	44
1980-81	526	16	29	39
1981-82	571	54	58	73
1982-83	526	14	20	37
1983-84	498	10	12	19
1984-85	647	47	70	96
1985-86	481	1	2	8
1986-87	459	3	7	10
1987-88	488	0	2	5
1988-89	509	4	3	13
1989-90	531	0	2	3
1990-91	503	3	2	4
1991-92	515	2	1	6
1992-93	499	0	5	3
1993-94	505	2	3	5
1994-95	500	1	2	3
1995-96	542	1	4	7
1996-97	575	2	5	4
1997-98	651	1	6	4
1998-99	680	3	3	7
1999-00	675	1	1	1
2000-01	672	0	0	2
2001-02	668	0	0	1
2002-03	661	0	1	2
2003-04	655	0	0	1
2004-05	635	0	0	1
2005-06	644	0	0	0
2006-07	648	0	0	0
2007-08	659	0	0	0
2008-09	643	0	0	0
2009-10	646	0	0	0
2010-11	656	0	0	0
2011-12	667	0	0	0
2012-13	706	0	3	2

Entrenadores

Temporada	Entrenadores	Sin Nombre Completo	Sin Datos	Sin Foto
1959-60	50	1	7	11
1960-61	52	0	2	10
1961-62	50	0	2	4
1962-63	48	0	4	6
1963-64	52	0	1	5
1964-65	49	0	2	6
1965-66	44	0	2	4
1966-67	51	1	1	1
1967-68	56	0	3	7
1968-69	36	0	3	1
1969-70	39	0	1	1
1970-71	42	0	2	0
1971-72	43	0	1	0
1972-73	30	0	0	0
1973-74	39	0	1	0
1974-75	40	0	3	3
1975-76	40	0	0	0
1976-77	32	0	1	0
1977-78	36	0	2	4
1978-79	29	0	0	0
1979-80	32	0	0	0
1980-81	29	0	0	0
1981-82	41	0	0	1
1982-83	30	0	0	1
1983-84	34	0	0	2
1984-85	30	0	0	0
1985-86	30	0	0	0
1986-87	29	0	0	0
1987-88	35	0	0	0
1988-89	34	0	0	1
1989-90	38	0	0	0
1990-91	35	0	0	0
1991-92	37	0	0	1
1992-93	42	0	1	4
1993-94	32	0	0	2
1994-95	45	0	3	4
1995-96	36	0	0	0
1996-97	47	0	0	0
1997-98	49	0	0	0
1998-99	47	0	0	0
1999-00	44	0	0	0
2000-01	38	0	0	0
2001-02	43	0	0	0
2002-03	40	0	0	0
2003-04	57	0	0	0
2004-05	40	0	0	0
2005-06	43	0	0	0
2006-07	43	0	0	0
2007-08	44	0	0	0
2008-09	40	0	0	0
2009-10	39	0	1	0
2010-11	39	0	0	0
2011-12	40	0	0	0
2012-13	32	0	0	0

Figura 12: Datos 2ª División

3.4.2. Extracción

Para llevar a cabo la extracción de la información de la web que albergaba los partidos[cita requerido], ha bastado con descargar todos los partidos en un fichero de texto con tamaño de campo fijo, el cual facilita su posterior procesado y almacenamiento en la base datos.

Sin embargo para la extraer la información de la web que contiene la mayor parte de los datos para nuestra base de conocimiento [7], ha sido necesario el uso de dos API's (*Application Programming Interface*), una dedicada a recorrer el sitio web y descargar las páginas HTML que necesitamos (*crawler* [8]) y otra que analiza el código HTML extrayendo la información deseada para su posterior almacenamiento (jsoup [3]).

La siguiente figura (figura 13) es una representación visual de como interaccionan las distintas herramientas utilizadas en esta fase del proyecto.

Figura 13: Esquema extracción y almacenamiento de informaci

Crawler

Un *crawler* es una herramienta capaz de recorrer de forma automática distintas páginas web sin necesidad de supervisión. Existen multitud de fines para aplicaciones de este tipo, desde indización de múltiples páginas web para los buscadores (googleboot, etc) hasta *spambots* capaces de saturar páginas web con publicidad.

La figura 14 representa un esquema conceptual de un *crawler.*

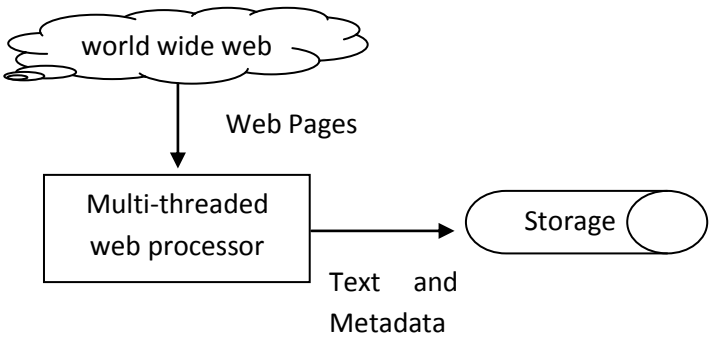

Figura 14: Esquema alto nivel web crawler standard

En nuestro caso usaremos un *crawler* para recorrer el sitio web www.bdfutbol.com, para descargar las páginas que contienen la información que necesitamos y poder procesarlas posteriormente. El *crawler* que hemos decidido utilizar es conocido como "**crawler4j**"[8], el motivo de utilizar este *crawler* es porque está implementado en Java, permitiéndonos incorporarlo a nuestra herramienta como una librería, porque posee una licencia apache que nos permite su uso, pero no su distribución comercial y por su simplicidad a la de configurarlo y manejarlo.

Para poder aprovechar al máximo el *crawler* en nuestro proyecto, hemos tenido que configurarlo, para ello hemos seguido las instrucciones que nos proporciona el sitio web, y hemos añadido dos clases dedicadas a su manejo, una encargada de almacenar los parámetros de configuración (clase *CrawlController*) y otra encargada de procesar la información (clase *Crawler*) ambas clases se encuentran dentro del paquete *crawler*.

Los parámetros de configuración y funciones que hemos creído relevantes han sido:

- Número de hilos de crawleado simultáneos
- Tiempo de espera entre solicitud de páginas web, este parámetro es importante si queremos evitar que nos confundan con un programa malicioso y el servidor nos añada a su lista negra de IP's.
- Profundidad máxima de crawleado, es el número máximo de niveles al que permitimos acceder al crawler para analizar.
- Página o páginas desde donde se iniciará el proceso de crawleado.
- Expresión regular que deben cumplir las direcciones web (*Uniform Resource Locator*, URL) detectadas para que el *crawler* las visite.
- Qué hacer cuando una url cumple nuestro patrón de filtrado.

Una vez configurado el *crawler*, se realizaron diversas pruebas para comprobar que los patrones de los filtros eran adecuados, la descarga de la información de las urls que visitaba y cómo podíamos diferenciar a través de la información que procesaba el *crawler* si una url pertenecía a un jugador, entrenador, etc.

Para comprobar la descarga de la información de las urls que se visitaban, hubo que sobrescribir el método "*visit*" de la api, de esta forma descubrimos que podíamos acceder de dos formas distintas al contenido de las urls que se visitaban:

a) Accediendo directamente al texto de la página, obteniendo directamente un string ya procesado por la api, método que se descartó por la imposibilidad de extraer de forma segura y eficiente los datos que requeríamos.
b) Obteniendo el código HTML de la página, tal y como lo haría un navegador. Este fue el método que decidimos desarrollar porque obteníamos toda la información necesaria dentro de un documento al que podíamos acceder de forma estructurada.

   Una vez descubierta la forma de acceder al contenido de las urls que el *crawler* visitaba, solo nos queda procesar el código HTML para extraer los datos que necesitamos; para esta tarea nos ayudamos de otra librería *opensource* llamada "jsoup". El proceso de extracción de la información lo explicamos más adelante, por lo que ahora no vamos a entrar en detalle.

Con respecto a los filtros de las urls, se diseñaron expresiones regulares que al combinarlas con los métodos de búsqueda de patrones de java, devolvían un valor booleano para saber si la url detectada se debía añadir al árbol de páginas para visitar del *crawler* o no y si se debía solicitar esa web o no. Las expresiones regulares utilizadas fueron las siguientes:

Expresión	Descripción
.	Indica cualquier carácter
^Expresión	El símbolo ^ indica el principio del String. En este caso el String debe contener la expresión al principio.
|	Operador OR
+	Indica 1 o más veces
*****	Indica 0 ó más veces
?	Indique 0 ó 1 vez
$	Indica el final de cadena de la cadena de texto

Tabla 23: Símbolos especiales expresiones regulares

a) Expresión que indica si una url encontrada en el código HTML de la página visitada se debe añadir al árbol de exploración del *crawler*, para ello nos fijamos en la terminación de la url, si no se corresponde con una terminación propia de un documento HTML o PHP no se visita.

.(.\\.(css|js|bmp|gif|jpe?g|png|tiff?|mid|mp2|mp3|mp4|wav|avi|mov|mpeg|ram|m4v|pdf|rm|smil|wmv|swf|wma|zip|rar|gz))$*

b) Expresión que controla el dominio de las urls a visitar, para ello se comprueba que el inicio de la url se corresponde con el dominio de la página, así nos aseguramos que nuestro *crawler* no abandonará el sitio web, evitando que se acceda a posibles enlaces externos del sitio y acelerando el proceso de crawleado.

*^http://www.bdfutbol.com/es/+.**

La ER comprueba que el inicio de la cadena de texto considerara como url es "*http://www.bdfutbol.com/es/*", seguido o no de cualquier otro carácter. Esta ER combinada con la explicada en el apartado a) mediante la puerta lógica AND, nos permite filtar qué urls debe tener en cuenta el *crawler* y cuáles no.

Por último para poder diferenciar entre las web que contienen la información que deseamos y saber si estamos accediendo a un jugador o a un entrenador, etc., probamos varios métodos:

a) Analizando la información disponible en los metadatos de las páginas web. Esta primera tentativa consistía en la búsqueda de palabras clave como "Jugador, Entrenador, Team, etc" entre las distintas etiquetas de metadatos de la web. En un principio este sistema parecía ser válido hasta que en un determinado momento el administrador del sitio web decidió cambiar la estructura de su web, modificando dichas etiquetas, entre otras cosas, lo que nos obligó a modificar nuestra estrategia de extracción.
b) La segunda estrategia que se llevó a cabo para la clasificación de las páginas web visitadas, fue analizando la url; observamos que la única diferencia en la url de dos jugadores era el

número identificativo de cada jugador, y que la diferencia entre url de la página de un jugador y otra distinta era que a partir de un determinado punto del dominio éste cambiaba.

Jugador 1 --> http://www.bdfutbol.com/es/j/j94.html
Jugador 2 --> http://www.bdfutbol.com/es/j/j358.html
Entrenador 1 --> http://www.bdfutbol.com/es/l/l1966.html

De esta forma somos capaces de identificar a qué tipo de página estamos accediendo con total seguridad.

Jsoup

Una vez seleccionada y descargada la página que contiene la información que necesitamos, tenemos que procesarla para poder extraer la información del documento DOM que deseamos incluir en nuestra base de datos. Para esta tarea hemos utilizado Jsoup [3], una herramienta diseñada especialmente para este tipo de situaciones y que además es compatible con JAVA, una de las ventajas de Jsoup es que es *opensource*, con licencia MIT que no ofrece ninguna restricción para su uso, modificación o distribución y que además podemos integrar perfectamente en nuestra herramienta.

Antes de empezar a analizar con Jsoup, debemos reconocer el tipo de url que vamos a analizar, para ello utilizamos la expresión regular que comentamos anteriormente, y dependiendo del tipo de url extraeremos y almacenaremos la información. Los tipos de url que encontraremos serán de Jugador, Entrenador y Equipo.

1. Si la url pertenece a un jugador, dividiremos la información a extraer en dos partes, la "ficha" y el "histórico". La información de la ficha contiene los datos de identificación del jugador, tales como su nombre, número identificador, etc., los cuales servirán para saber quién es cada jugador y se almacenará en la tabla "Jugadores" de la base de datos:

   Mientras que el histórico del jugador, se almacenará en la tabla "Historico_jugador" de la base de datos, la cual estará relacionada mediante el identificador único de cada jugador con la tabla que contiene la información del jugador.

2. Cuando la dirección web pertenezca a un entrenador la extracción de información será similar a la del jugador, ya que la página del entrenador reutiliza los nombres de las etiquetas de los objetos DOM de los jugadores. De esta forma extraeremos la información del entrenador y lo almacenaremos en la tabla entrenadores de la bbdd y el histórico del entrenador lo almacenaremos en la tabla "Trayectoria_Entrenador", relacionando ambas por el identificador único del entrenador, que al ser único, nos permite comprobar si el entrenador es un ex-jugador cuando encontramos coincidencias en la tabla "Jugadores"

3. Por último si la url encontrada pertenece a un equipo, extraeremos toda la información perteneciente a la ficha del equipo y la almacenaremos en la base de datos con un identificador único. En caso de equipos que sean refundados los almacenaremos en la

BBDD con un identificador nuevo, pero crearemos el campo "Alias" para almacenar el nombre de los equipos anteriores ya que el resto de información que ofrece la página será la mezcla del equipo con la nomenclatura anterior y la nueva.

Además de esto, tenemos que tener en cuenta que los nombres de los equipos almacenados en la otra página web que utilizaremos para extraer los partidos, pueden no ser nombrados igual que en esta página web. Para solucionar esta dificultad, debemos asignar un número identificativo único a los equipos y realizar algún procedimiento automático para correlacionar los nombres de los equipos de una base de datos y otra, además debemos tener en cuenta que el método que desarrollemos debería poder reutilizarse para cuando añadamos los partidos a predecir, ya que los nombres de los equipos que se utilicen podrían no corresponderse con los ya almacenados en la BBDD. Para este problema hemos desarrollado la siguiente propuesta:

En primer lugar hemos creado un fichero de lenguaje de marcas extensible (*eXtensible Markup Language*, XML), que nos permite utilizar el estándar DOM que ya conocemos para recorrerlo, que almacenará en estructuras jerárquicas el nombre del equipo y su correspondiente identificador único en la base de datos, de esta forma cuando recibamos un equipo comprobaremos el fichero XML y extraeremos el id que usaremos en nuestro sistema. La estructura que almacenará el alias del equipo será la siguiente:

<team>
<id>266</id> --> Identificador único del equipo en la BBDD
<name>premià</name> --> Nombre del equipo
</team>

En segundo lugar hemos utilizado la distancia Levenshtein [19] para los nuevos alias que no tengamos controlados, de esta forma cuando un nuevo alias aparezca y no éste no esté recogido ni en la BBDD ni en el fichero XML, la distancia Levenshtein nos indicará qué alias de equipo ya recogido es el que más se le parece y lo trataremos como tal, añadiendo este nuevo alias a la BBDD y al fichero. De esta forma automatizamos el proceso de renombrado de equipos o nuevos alias, aunque hay que tener en cuenta que éste método no es a prueba de fallos, por lo que es necesario una revisión periódica de las nuevas incorporaciones, para asegurarnos que las desambiguaciones se hacen correctamente, i.e:

Un posible alias del Real Madrid C.F. podría ser el siguiente:

```
<team>
  <id>2</id>
  <name>real madrid</name>
</team>
```

Y un posible alias del Atlético de Madrid podría ser el siguiente:

```
<team>
  <id>7</id>
  <name>at. madrid</name>
</team>
```

En el hipotético caso en el que nos encontrásemos como un nombre de equipo el alias "madrid", nuestro método diría que el equipo más probable para este alias sería el Atlético de Madrid, ya que la distancia de Levenshtein sería menor que para real madrid, sin embargo nosotros sabemos que el alias madrid se suele utilizar para referirse al Real Madrid C.F., así que no nos queda más remedio que añadir manualmente este alias a nuestro fichero de la siguiente forma, asegurándonos que la próxima vez que aparezca ese término se desambigüe correctamente:

```
<team>
  <id>2</id>
  <name>madrid</name>
</team>
```

El problema para relacionar este tipo de situaciones no es fácil, ya que por norma general los procesos automatizados para desambiguar palabras requieren de un contexto para realizar la tarea con cierta fiabilidad, y aún así no se asegura que sea exacta, sin embargo en nuestro caso este sistema no podemos aplicarlo porque carecemos de contexto alguno, y un fallo en este asunto puede condicionar mucho las predicciones. Así que se recomiendan revisiones manuales de las nuevas incorporaciones para asegurar el correcto funcionamiento del sistema.

3.4.3. Almacenamiento

Para almacenar al base de conocimiento de una forma en la que se conserve íntegra y sea posible un acceso rápido y ordenado, hemos recurrido al sistema MySQL porque cumple todos los requisitos que necesitamos, es gratuito, y se puede integrar fácilmente con nuestra aplicación desarrollada en Java. El nombre de nuestra base de datos relacional será "Liga", porque representa bastante bien la información que va a contener y es intuitivo.

Se ha optado por un sistema de almacenamiento relacional o "tradicional" frente a los nuevos sistemas de almacenamiento no-relacional, porque la información a almacenar no es lo suficientemente grande como para requerir de estos nuevos sistemas de almacenamiento y porque un sistema de almacenamiento relacional es lo bastante robusto y escalable para lo que necesitamos, ya que no vemos necesidad de utilizar un sistema no-relacional al cual apenas se le va a sacar provecho y además en nuestro caso en concreto los rendimientos comparados de ambos sistemas serán muy parejos debido al volumen de datos que vamos a manejar.

La figura 15 muestra un diagrama de la base de datos diseñada para el proyecto:

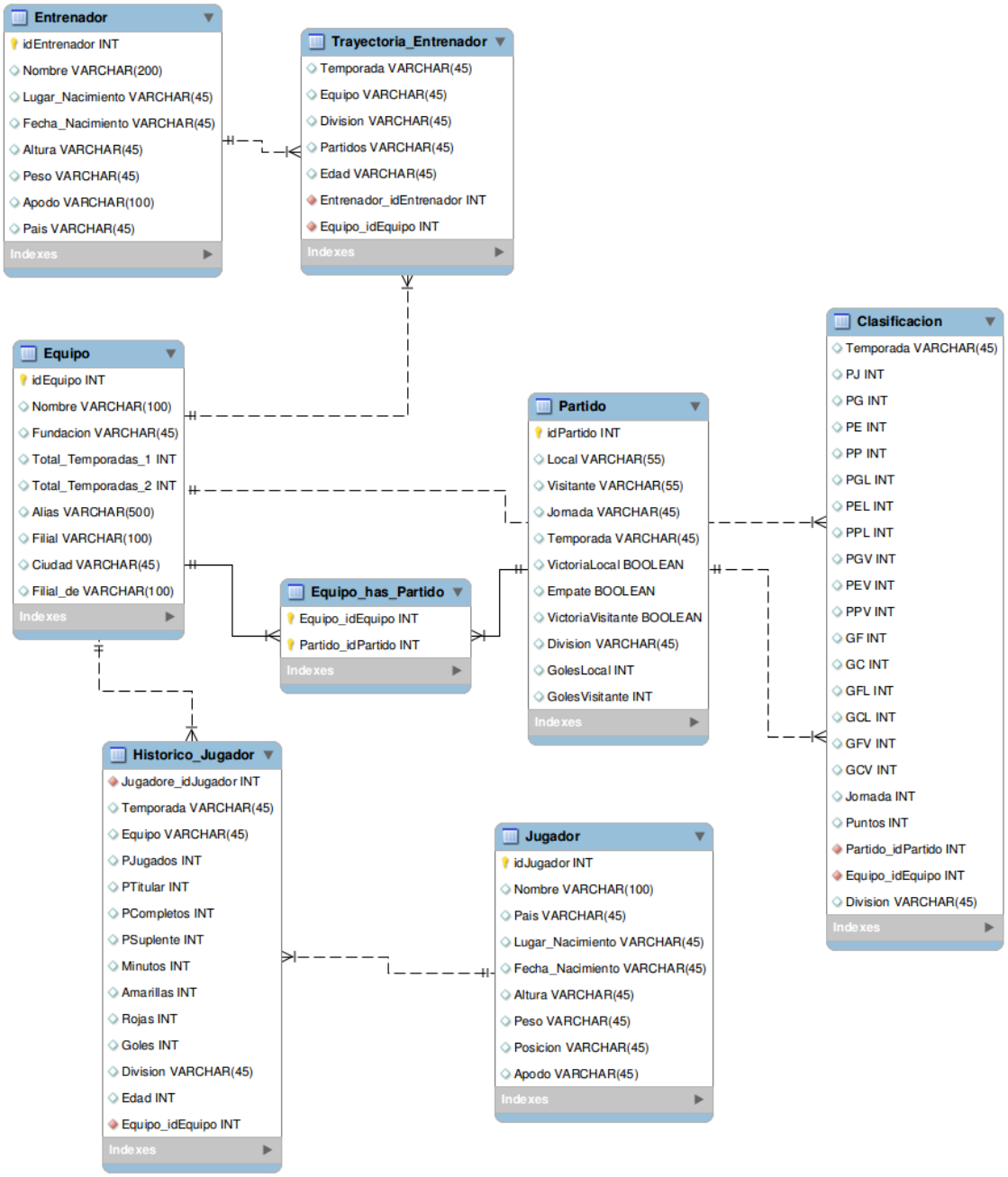

Figura 15: Modelo entidad relación de la BBDD

Tablas

Las tablas necesarias para almacenar la información son las siguientes:

- Equipo: Esta tabla está destinada a guardarla información que se ha creído relevante de los equipos. Contiene los siguientes campos:
    - IdEquipo: Identificador único de equipo
    - Nombre: Nombre del equipo
    - Fundación: Fecha de fundación del equipo cuando esté disponible.
    - Total_Temporadas_1: Nº total de temporadas en 1ª división del equipo
    - Total_Temporadas_2: Nº total de temporadas en 2ª división del equipo
    - Alias: Lista de pseudónimos del equipo separados por coma.
    - Filial: Valor booleano que indica si el equipo posee equipo filial.
    - Ciudad: Localización del equipo
    - Filial_de: Indica si el equipo es filial de algún equipo, en caso de serlo se indica el nombre del equipo del que es filial.

- Entrenador: Tabla dedicada a los entrenadores de que han pasado por la liga de fútbol profesional española; contiene los siguientes campos:
    - idEntrenador: Identificador único del entrenador, se puede corresponder con el identificador de un jugador, en cuyo caso eso nos indica que el entrenador ha sido previamente jugador de fútbol profesional.
    - Nombre: Nombre del entrenador.
    - Lugar_Nacimiento: Lugar de nacimiento del entrenador, cuando se disponga.
    - Fecha_Nacimiento: Fecha de nacimiento del entrenador, cuando se disponga.
    - Altura: Altura del entrenador, cuando se disponga.
    - Peso: Peso del entrenador, cuando se disponga.
    - Apodo: Mote del entrenador.
    - País: País de nacimiento del entrenador.

- Trayectoria Entrenador: Esta tabla se encarga de almacenar el historial de equipos por los que ha pasado el entrenador.
    - idEntrenador: Identificador del entrenador.
    - idEquipo: Identificador del equipo entrenado.
    - Temporada: Temporada que estuvo entrenando al equipo
    - Partidos: El número de partidos que el entrenador entrenó al equipo durante esa temporada.
    - División: División en la que se encontraba el equipo cuando lo entrenó.
    - Equipo: Nombre del equipo
    - Edad: Edad del entrenador

- Jugador: Tabla que almacenará la información relevante del jugador. Los campos que contiene son:
    - idJugador: Identificador único del jugador, se puede corresponder con el identificador de un entrenador, en cuyo caso significaría que dicho jugador se convirtió en entrenador o fue entrenador antes de ser jugador.
    - Nombre: Nombre del jugador.
    - País: Nacionalidad del jugador.
    - Lugar_Nacimiento: Lugar de nacimiento del jugador.
    - Fecha_Nacimiento: Fecha de nacimiento del jugdor.
    - Altura: Altura del jugador.
    - Peso: Peso del jugador.
    - Posición: Área del campo en la que juega el jugador.
    - Apodo: Mote o pseudónimo del jugador.

- Histórico_Jugador: Tabla que contiene los equipos por los que ha pasado el jugador, con estadísticas de su paso por ese equipo, esta tabla contiene un registro por cada temporada que el jugador ha estado en activo. Contiene los siguientes campos:
    - idJugador: Identificador del jugador.
    - Temporada: Temporada en la que jugó el jugador en ese equipo.
    - Equipo: Equipo en el que jugó el jugador.
    - idEquipo: Identificador del equipo en el que jugó el jugador.
    - PJugados: Partidos que jugó el jugador la temporada correspondiente en el equipo indicado.
    - PTitular: Partidos que el jugador jugó como titular en el equipo dicha temporada.
    - PCompletados: Partidos completos que el jugador jugó en el equipo la temporada correspondiente.
    - PSuplente: Partidos del equipo que inició el jugador como suplente y acabó en el campo.
    - Minutos: Minutos totales que jugó durante la temporada en el equipo.
    - Amarillas: Tarjetas amarillas que recibió el jugador durante la temporada
    - Rojas: Tarjetas rojas que recibió el jugador durante la temporada, tanto por doble amarilla como roja directa.
    - Goles: Goles a favor de su equipo anotados por el jugador.
    - División: División a la que pertenecía el equipo en el que jugó.
    - Edad: Edad del jugador en esa temporada.
    - Valoración: Valoración del jugador en cada temporada, basada en los datos estadísticos disponibles en la tabla.

- Partido: Tabla dedicada a almacenar la información relacionada de cada partido jugado en primera o segunda división desde la temporada 1972-1973. Los campos que contiene son:
    - idPartido: Identificador único del partido.
    - Local: Identificador del equipo local
    - Visitante: Identificador del equipo visitante.
    - Jornada: Jornada de disputa del partido.

  - Temporada: Temporada en la que se disputó el partido.
  - Victoria_Local: Booleano que indica si el partido acabó con victoria local.
  - Victoria_Visitante: Booleano que indica si el partido acabó con derrota local.
  - Empate: Booleano que indica si el partido acabó en empate.
  - División: División en la que se tuvo lugar el enfrentamiento (1º, 2º, Promoción)
  - Goles_Local: Número de goles anotados por el equipo local en el partido.
  - Goles_Visitante: Número de goles anotados por el quipo visitante en el partido.

- Clasificación: Tabla resumen de la clasificación hasta la fecha indicada del equipo. Esta tabla sirve para resumir la trayectoria general del equipo durante las distintas jornadas de una determinada temporada. Para generar esta tabla se ha recurrido a la tabla Partidos, de esta forma podemos guardar en un formato altamente accesible la información relacionada con un equipo a lo largo de una temporada. Contiene los siguientes campos:
  - Temporada: Temporada resumen del equipo.
  - División: División en la que se encuentra el Equipo
  - Jornada: Jornada hasta la que se ha calculado el resumen.
  - idEquipo: Identificador único del equipo.
  - idPartido: Identificador único del partido jugado en esa jornada de esa temporada por el equipo.
  - Puntos: Puntos acumulados por el equipo hasta la fecha en la temporada indicada.
  - PJ: Partidos jugados por el equipo hasta la fecha.
  - PG: Acumulado de partidos ganados por el equipo hasta el momento.
  - PP: Acumulado de partidos perdidos por el equipo.
  - PE: Acumulado de partidos Empatados por el equipo.
  - PGL: Acumulado de partidos ganados como local por el equipo.
  - PPL: Acumulado de partidos perdidos como local por el equipo.
  - PEL: Acumulado de partidos empatados como local por el equipo.
  - PGV: Acumulado de partidos ganados como visitante por el equipo.
  - PPV: Acumulado de partidos perdidos como visitante por el equipo.
  - PEV: Acumulado de partidos empatados como visitante por el equipo.
  - GF: Acumulado de goles a favor por el equipo.
  - GC: Acumulado de goles en contra por el equipo.
  - GFL: Acumulado de goles a favor por el equipo.
  - GCL: Acumulado de goles en contra por el equipo.
  - GFV: Acumulado de goles a favor como visitante por el equipo.
  - GCV: Acumulado de goles en contra como visitante por el equipo.

Relaciones

Para poder almacenar la información en la base de datos de forma legible, y que ésta mantenga la integridad, hemos utilizado claves primarias o índices en aquellas tablas que hemos considerado críticas, así como relaciones entre tablas para asegurar la integridad de la base datos. A continuación explicamos y detallamos dichas claves primarias y relaciones:

Índices

- idEntrenador*: Índice numérico y único para identificar a cada uno de los entrenadores almacenados en la base de datos, además si el entrenador fue anteriormente jugador el valor del índice se conservará.
- idEquipo: Valor numérico único para identificar a cada uno de los equipos almacenados.
- idPartido: Valor numérico utilizado para identificar cada uno de los partidos almacenados en la base de datos, además este índice es autoincrementado, por lo que cuando se añada un nuevo partido a la base de datos, ésta se encargará de asignarle valor al índice.
- idJugador*: Valor numérico utilizado para identificar a cada uno de los jugadores contenidos en la base de datos, además si posteriormente este jugador se convierte en entrenador, el valor del índice se conservará y pasará a ser su idEntrenador.

  *El motivo de utilizar el mismo índice para un jugador que luego pasa a ser entrenador o viceversa, viene heredado del sitio web del que se obtiene la información de los jugadores y entrenadores.

Relaciones

Relaciones 1:N

Este tipo de relaciones crea una relación de un elemento con varios, de forma que un elemento de una tabla puede estar relacionado con varios elementos de una segunda tabla, pero cada registro de la segunda tabla sólo puede estar relacionado con 1 elemento de la primera tabla; suele ser la relación más común.

En nuestro caso tenemos las siguientes relaciones de este tipo:

a. Jugador-Histórico_Jugador: Se ha creído necesario establecer este tipo de relación para poder identificar la trayectoria de un jugador mientras ha estado en activo.
b. Equipo-Histórico_Jugador: Esta relación es necesaria para poder identificar por los equipos que ha pasado un jugador.
c. Equipo-Clasificación: La relación ayuda a relacionar al información contenida en la tabla clasificación, con su respectivo equipo.
d. Equipo-Trayectoria_Entrenador: Al igual que la relación de Equipo-Histórico_Jugador, esta relación permite conectar un equipo con los entrenadores que ha tenido a lo largo de su historia.
e. Entrenador-Trayectoria_Entrenador: Esta relación realiza una función similar a la de Jugador-Histórico_Jugador, permitiendo relacionar la trayectoria de un entrenador con su entrenador correspondiente.
f. Partido_Clasificación: Esta relación ayuda a establecer una relación entre un partido, y la clasificación de los equipos que juegan ese partido.

Relaciones N:M:

Estas relaciones permiten que los registros de una tabla puedan estar enlazados con varios registros de una segunda tabla y viceversa. Para poder realizar este tipo de relaciones es recomendable generar una tabla intermedia que se relaciona mediante relaciones 1-N con las tablas implicadas en la relación. De esta forma se evita la repetición de registros en las tablas implicadas.

En nuestro caso solamente disponemos de una relación de este tipo:

I. Equipo-Partidos: El motivo de utilizar esta relación, no es otro que el de poder identificar a los rivales de un partido, y como en cada partido existen dos rivales, no queda más remedio que utilizar una relación de este tipo para evitar repetición de registros.

3.5.Procesado y predicción

Como hemos explicado anteriormente la base de datos se nutre de dos fuentes independientes. La información extraída de cada una de las fuentes de información se ha almacenado correctamente en la BBDD, proporcionando una gran cantidad de datos que podremos analizar y moldear para extraer el mayor conocimiento posible para nuestro predictor.

3.5.1. Procesado

En sintonía con la labor de encontrar un modelo de predicción óptimo, se ha creído conveniente hacer un procesado previo de la información contenida en la BBDD, para de esta forma agilizar el cálculo de algunos parámetros que utilizaremos más adelante y así parametrizar o asignar un valor numérico a algunas relaciones nuevas que creemos que pueden ser relevantes para nuestro modelo.

a. Creación de la tabla "Clasificación":

La tabla clasificación se ha generado a partir de los datos almacenados en la tabla "Partidos". La tabla partidos contiene todos los partidos de la Liga BBVA y de la Liga Adelante desde la temporada 1972-73, almacenando en cada registro la temporada, jornada, división, rivales y resultado de cada uno de los partidos disputados. En sí la tabla "Partidos" alberga potencialmente mucha información, pero si la dejamos almacenada de ese modo no podemos extraer mucha información útil para nuestro predictor, por lo que tenemos que procesarla para que esa información latente que posee, salga a la luz y podamos aprovecharla.

El procesado de la tabla partidos consistirá en generar una tabla nueva que llamaremos "Clasificación", la cual albergará datos estadísticos extraídos de los partidos. En la tabla "Clasificación" se generará un registro para cada equipo que haya disputado un partido durante las jornadas pertenecientes a cada una de las temporadas disponibles en la tabla "Partido". Dichos registros contendrán varios parámetros que nos ayudarán a visualizar la evolución de un equipo durante la temporada.

Los parámetros calculados ya han sido explicados en el apartado anterior

Tablas, dentro de este mismo punto, por lo que no vamos a volver a explicar cada uno de los parámetros de nuevo.

b. Algoritmo de valoración de jugadores:

Además de la creación de la tabla "Clasificación", hemos creído que a la información extraída del histórico de un jugador se le podía sacar algo más de rendimiento, por lo que hemos desarrollado un algoritmo que creemos es capaz de valorar a un jugador en base a los datos que nos ofrece la página web sobre cada jugador.

Este algoritmo está diseñado de forma que no importa el número de atributos que se vayan a utilizar para la valoración del jugador, aún así ese parámetro los vamos a mantener constante para que la valoración de los jugadores sea equitativa. Este coeficiente que vamos a calcular representa el área de un polígono irregular inscrito en una circunferencia.

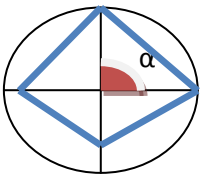

Figura 16: Polígono irregular inscrito

La idea para calcular el área del polígono consiste en usar la técnica de triangulación de un polígono en nuestro favor, para ello vamos a dividir la circunferencia en la que está inscrito el polígono en partes iguales, tantas como atributos tenemos, de esta forma mantenemos el ángulo comprendido entre dos subdivisiones contiguas constantes, después la longitud del radio será el valor del atributo, de esta forma disponemos de dos lados contiguos y el ángulo formado por ellos, información suficiente para calcular el área de un triángulo, por lo que solo nos queda calcular el área de cada uno de los triángulos formados y sumarla.

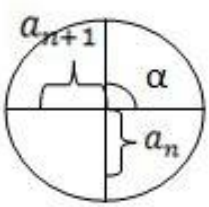

Figura 17: Representación parámetros

Por lo tanto fórmula utilizada para calcular la valoración de los jugadores es la siguiente, siendo n el número de atributos utilizados y a_n el valor del atributo:

$$A = \frac{1}{2} \times \sin\alpha \sum_{n=0}^{n+1} a_n \times a_{n+1}$$

α será el ángulo interior de cada una de las partes en las que dividiremos a circunferencia así: $\alpha = {}^{360}/_{n}$

Después de aplicar el algoritmo en nuestra base de datos y almacenar los valores resultantes en la columna "Valoración" de la tabla "Histórico_Jugador", hemos querido comprobar si el algoritmo funciona correctamente, para lo cual hemos ejecutado la siguiente consulta:

SELECT h.Jugador_idJugador as Jugador, h.Temporada as Temporada, j.Nombre as Nombre, h.Valoracion as Valoracion FROM Historico_Jugador h JOIN Jugador j ON h.Jugador_idJugador= j.idJugador ORDER BY Valoracion DESC

que devuelve el nombre del jugador, su identificador único, la valoración del jugador y la temporada en la que se consiguió esa valoración, ordenando los registros de respuesta según su valoración, de mayor a menor. El orden lo hemos impuesto así para comprobar si los mejores jugadores que han pasado tanto por La Liga BBVA como por la Liga Adelante aparecen en las primeras posiciones, sino es así es porque el algoritmo es erróneo y no nos sirve:

#	Jugador	Temporada	Nombre	Valoracion
1	1753	2011-12	Lionel Andrés Messi Cuccittini	1371.9526055810773
2	12429	2011-12	Cristiano Ronaldo Dos Santos Aveiro	1313.7337769505539
3	306	1986-87	Baltazar María De Moráis Junior	1272.5843431017547
4	1040	1986-87	Hugo Sánchez Márquez	1161.0236557783433
5	1753	2012-13	Lionel Andrés Messi Cuccittini	1122.3127068983617

Figura 18: Valoración jugadores

En la figura podemos apreciar que las primeras posiciones están copadas por delanteros, esto es debido a que la información almacenada de los jugadores para cada temporada es bastante limitada, y el atributo que marca la diferencia en este caso es el número de goles, por lo que es lógico que las primeras posiciones las ocupen los jugadores más goleadores.

Este hecho condicionará sobremanera los valores de esta columna, pero aún así creemos que suficientemente representativo, ya que los partidos se ganan marcando goles y los equipos con los mejores goleadores suelen ganar.

3.5.2. Weka

WEKA (*Waikato Environment for Knowledge Analysis*) es una herramienta desarrollada por la Universidad de Waikato en Nueva Zelanda. El desarrollo de esta herramienta empezó en 1993, con el propósito de crear un software para analizar información sobre los cultivos de la zona y mejorar las producciones; esta primera versión de WEKA se codificaría en C.

Unos años más tarde se decidió reescribir todo el código de la herramienta a JAVA, un lenguaje orientado a objetos que empezaba a cobrar fuerza en aquella época y además se actualizó el código con varias implementaciones nuevas de algoritmos de modelado.

A partir de ese momento WEKA empezó a ganar fama entre los investigadores en el campo de la inteligencia artificial por recibir prestigiosos galardones y porque WEKA permitía a los investigadores incorporar algoritmos propios al núcleo de la herramienta, además WEKA facilitaba la integración con otras aplicaciones al poder utilizarse como librería, factor que otras herramientas del área no permiten, como el simulador de redes neuronales de la universidad de Stuttgart (SNNS) o la herramienta especializada en técnicas de agrupamiento Elki.

Para nuestro proyecto usaremos la versión más reciente de WEKA, concretamente la versión de desarrollo 3-7-11. Hemos seleccionado la versión de desarrollo de la herramienta frente a la versión estable, porque ésta permite incorporar en WEKA, a través de la interfaz gráfica, algoritmos propios.

Entre todas las funcionalidades que ofrece WEKA, nuestro proyecto utilizará, principalmente, tres de ellas, pre-procesado, clasificación y selección de atributos:

a. **Pre-procesado:** Permite aplicar filtros sobre los conjuntos de datos a utilizar, además la interfaz gráfica muestra estadísticas relacionadas con el conjunto de datos cargado.

b. **Clasificación:** Esta funcionalidad de la herramienta permite utilizar cualquiera de los algoritmos de clasificación pre-configurados o cargados por nosotros mismo, en nuestro caso esta funcionalidad es clave porque nos servirá para generar nuestro modelo de predicción.

c. **Selección de atributos:** Gracias a esta funcionalidad de WEKA, podemos ver la relevancia que poseen los distintos atributos de un conjunto de datos sobre el resultado final. Esta funcionalidad nos permitirá evaluar los atributos que utilicemos o transformarlos, dependiendo del algoritmo que utilicemos.

4. Modelo predicción

En la actualidad existe una gran variedad de sistemas de predicción: sistemas basados en reglas, basados en el teorema de Bayes, en redes neuronales, y un largo etcétera de sistemas con características propias.

En este apartado explicaremos las distintas técnicas y procedimientos utilizados para generar nuestro modelo de predicción.

4.1.Perceptrón multicapa (MLP)

Un perceptrón tiene dos acepciones válidas, la primera y probablemente para la cual se acuñó el término, se refiere a un algoritmo capaz de generar un criterio para seleccionar elementos de una colección, mientras que la otra acepción se refiere a un tipo de red neuronal artificial. El problema o limitación de este elemento es que en 1969 se demuestra en un artículo que un perceptrón simple no puede resolver problemas no lineales, sin embargo si se utilizan varios de estos perceptrones en común, sí es posible, al menos teóricamente, la resolución de problemas no lineales.

En la figura 19 se muestra un diagrama de un perceptrón simple.

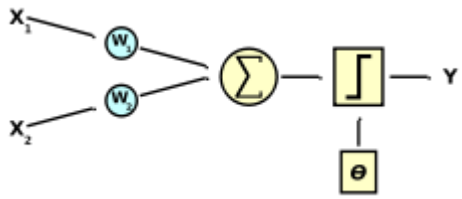

Figura 19: Computer.Science.AI.Neuron (*wikipedia commons*)

Esta limitación redujo el entusiasmo de los investigadores por este tipo de tecnología, quedando relegada a un segundo plano en el ámbito de la I.A. hasta 1986, cuando se presenta la Regla Delta Generalizada o algoritmo de *backpropagation*, la cual permite interconectar multitud de perceptrones simples formando una red, para la resolución de problemas no lineales.

La interconexión, parcial o completa, de varios perceptrones simples da lugar a las redes neuronales. Una red neuronal artificial (RNA o ANN) es una propuesta tecnológica relacionada con el aprendizaje y el procesamiento automático inspirado por el sistema nervioso de los seres humanos. Este sistema consiste en una interconexión de neuronas que cooperan entre sí produciendo respuestas a los estímulos recibidos.

La figura 20 representa el concepto de similitud entre red neuronal artificial, y el sistema nervioso central:

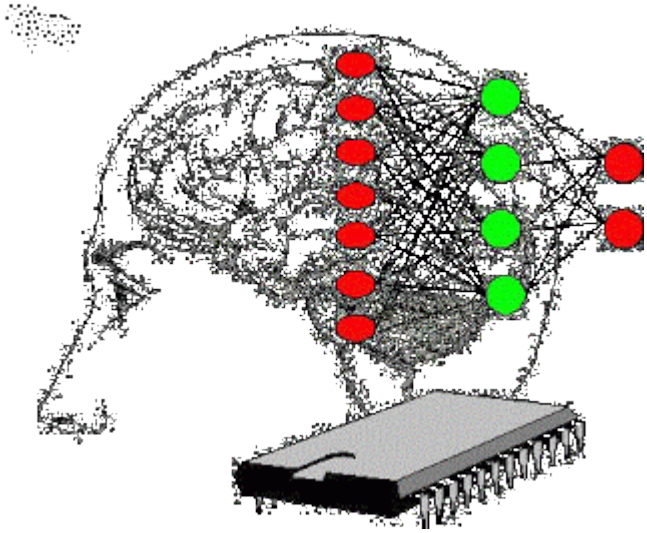

Figura 20: Representación de una red neuronal

Las salidas generadas por las distintas neuronas que componen la red depende de varias funciones:

1. **Función de excitación**, que consiste en calcular el valor de entrada de la neurona, el cual depende de los valores de salida de las neuronas conectadas con ella y del valor del peso de los enlaces que las conectan.
2. **Función de activación**, la cual modificará el valor devuelto por la función de excitación si el valor devuelto cumple las condiciones necesarias para activar dicha función.
3. **Función de transferencia**, que acota la salida de la neurona según la interpretación que deseemos darle a las salidas de la red.

Para nuestro proyecto vamos a utilizar una red neuronal de multiples capas, más conocido como perceptrón multicapa (*multi-layer perceptron*, MLP). Como ya hemos explicado las redes neuronales permiten solventar problemas no lineales y esta característica se mantiene aún combinando varias redes neuronales en un mismo sistema.

Cada una de las redes neuronales que componen un perceptrón multicapa se conoce como capa, y dependiendo de la funcionalidad a la que estén destinadas esas capas, podemos clasificarlas como:

- Capa de entrada: Son las neuronas que introducen los patrones en la red en crudo, o sea sin procesado.
- Capa Oculta: Son aquellas redes de neuronas intermedias entre la capa de entrada y la de salida, es en estas capas donde se procesa el problema.
- Capa de salida: Son las neuronas cuyas salidas se corresponden con las salidas de la red.

En la figura 21 podemos observar un diagrama de un MLP

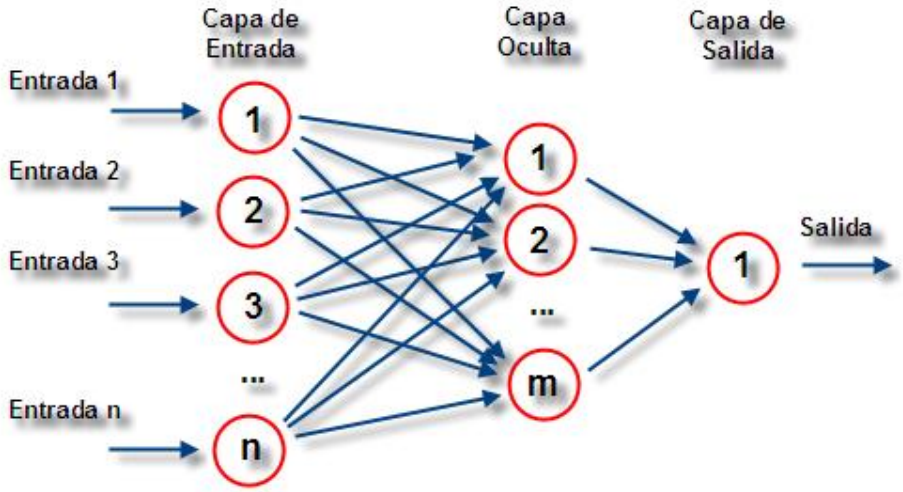

Figura 21: Perceptrón multicapa (*Wikipedia Commons*)

Los MLP dependiendo de la interconexión de sus capas ocultas pueden clasificarse como MLP totalmente conectados o parcialmente conectados, en el caso de que un MLP esté totalmente conectado (como el de la figura anterior) significa que cada una de las neuronas de una capa está conectada con todas las neuronas de la capa inmediatamente posterior, mientras que un MLP parcialmente conectado no conecta todas las neuronas de una capa con todas las de la capa posterior, sino que solamente las conecta con una parte, formando estructuras modulares que se pueden conectar entre sí dando lugar a redes neuronales mayores, que dependiendo de la forma en que se conecten estas estructuras podrían clasificarse como estructuras en paralelo las cuales deciden las soluciones de la red de forma equitativa o jerárquicas**¡Error! No se encuentra el origen de la referencia.**.

A continuación se muestra una representación gráfica (figura 22) de una red neuronal parcialmente conectada, con una estructura en paralelo.

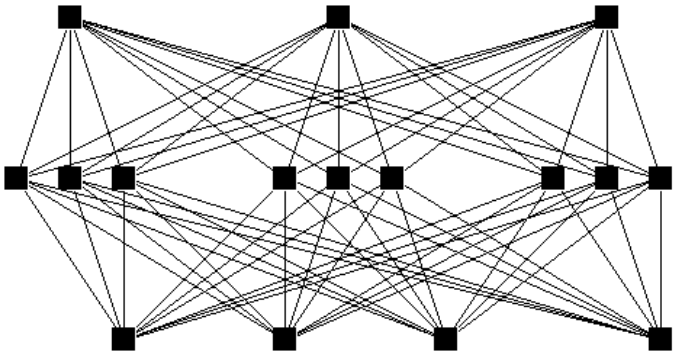

Figura 22: Red neuronal parcialmente conectada

El MLP es un modelo altamente contrastado y que puede ser utilizado en problemas como el que tratamos de resolver, por lo que vamos a centrar nuestras pruebas en analizar el comportamiento de un MLP en la labor de predicción de los partidos de la Liga BBVA y la Liga Adelante.

4.1.1. Algoritmo aprendizaje

Existen numerosos algoritmos de aprendizaje que se pueden aplicar a una red neuronal, como pueden ser el *Backpropagation*, *Quickpropagation*, *Delta Rule*, etc. En la práctica realizada anteriormente por el autor de este proyecto y sus compañeros probaron varios de estos algoritmos, todos ellos ya estaban implementados en la herramienta JavaSNNS, y llegaron a la conclusión de que el algoritmo más apropiado para este problema era el *Backpropagation*, además WEKA solo incorpora este algoritmo en su MLP, por lo que de esta forma nos ahorramos tener que desarrollar el algoritmo nosotros mismos.

BackPropagation

El algortimo de entrenamiento *Backpropagation* es probablemente el más usado. Este algoritmo pertenece a la rama de algoritmos de aprendizaje supervisado, y su funcionamiento consiste en ajustar los pesos de las conexiones de las neuronas tratando de minimizar el error cuadrático.

Para poder aplicar *Backpropagation* en una red neuronal, debemos utilizar una red neuronal inicializada, esto significa que todos los pesos de la red poseen unos valores iniciales aleatorios. Estos valores iniciales se irán modificando a medida que la red aprenda.

El proceso de aprendizaje es iterativo y se realiza desde la última capa de neuronas hacia la primera según los patrones o instancias van pasando por la red. Debido a que el aprendizaje se fundamenta en la repetición, es necesario disponer de un conjunto de datos de entrenamiento amplio, que además tendremos que pasárselo a la red varias veces para que ésta aprenda, por lo que es un proceso cíclico que se repetirá tantas veces como sea crea necesario, ya que no existe a priori un número predeterminado de ciclos que nos indique que una red ya ha aprendido, sino que dependerá de cada red. A este ciclo de aprendizaje de una red neuronal o MLP lo llamaremos *época.*

El número de iteraciones del algoritmo *backpropagation* equivale al número de épocas multiplicado por el número de instancias o patrones pertenecientes al conjunto de entrenamiento. Podemos identificar cuatro fases o hitos en cada iteración de aprendizaje:

1. La red recibe una instancia de datos y se calcula su salida.
2. Se comprueba si la salida ha sido correcta, sino lo ha sido se determina el error.
3. Se determina la cantidad en que se debe modificar cada peso
4. Se actualiza el valor de los pesos

Para poder utilizar correctamente este algoritmo de aprendizaje en nuestra red neuronal, tenemos que conocer sus posibles puntos flacos: Como ya sabemos podemos condicionar la velocidad con la que aprende una red neuronal modificando su coeficiente de aprendizaje, por lo tanto debemos tener en cuenta que un coeficiente de aprendizaje muy pequeño hace que el entrenamiento sea demasiado o que incluso se estanque antes de tiempo; pero si nos pasamos con el valor de la tasa de aprendizaje podemos hacer que el algoritmo se vuelva inestable haciendo que oscile muy deprisa. No existe una solución estándar para este problema, ya que el entrenamiento de una red con este algoritmo es un ejercicio de prueba y error que no nos asegura alcanzar un error mínimo global, sino que puede ser mínimo local, además está la dificultad añadida de que si

nos pasamos con el número de épocas podemos hacer que la red se sobre-adecúe a conjunto de entrenamiento y no nos sea de utilidad para datos o instancias nuevos.

Por este motivo hemos desarrollado un plan de pruebas lo bastante amplio como para poder evitar y detectar estos problemas cuando aparezcan.

4.1.2. Entradas

Llegados a este punto creemos conveniente hacer una aclaración a la nomenclatura que utilizaremos a partir de ahora:

- Instancia: Patrón de datos correspondiente a un enfrentamiento.
- Parámetro: Atributo perteneciente a una instancia
- MLP: Perceptrón multicapa
- Conjunto de entrenamiento: Grupo de instancias encargado de crear el modelo.
- Conjunto de test: Grupo de instancias utilizadas para comprobar la eficacia del modelo.

Para poder aprovechar al máximo nuestra base de datos y pasar a la red neuronal la mayor cantidad de información posible que nos ayude a predecir el resultado de un partido, hemos utilizado los siguientes parámetros:

1. idTeam1: Identificador utilizado en la base de datos para identificar al equipo local
2. idTeam2: Identificador utilizado en la base de datos para identificar al equipo visitante.
3. percent-match-local-win: Porcentaje de partidos que ha ganado el equipo que jugaba como local cuando se han enfrentado estos equipos, sin diferenciar donde se ha jugado el partido.
4. percent-match-local-defeat: Porcentaje de partidos que ha ganado el equipo visitante cuando se han enfrentado estos equipos, sin distinciones de campo.
5. percent-match-draw: Porcentaje de partidos disputado entre los dos equipos que ha acabado con empate.
6. percent-team1-win: Porcentaje de partidos disputado entre los dos equipos que ha ganado el equipo1.
7. percent-team2-win: Porcentaje de partidos disputado entre los dos equipos que ha ganado el equipo2.
8. percent-team1-seasonWins: Porcentaje de partidos que ha ganado el equipo1 durante la temporada.
9. percent-team1-seasonLocalWins: Porcentaje de partidos que ha ganado el equipo1 durante la temporada jugando como local.
10. percent-team1-seasonLocalDraws: Porcentaje de partidos que ha empatado el equipo1 durante la temporada jugando como local.
11. percent-team1-seasonLocalDefeats: Porcentaje de partidos que ha perdido el equipo1 durante la temporada jugando como local.
12. team1-average: Media de partidos ganados frente a jugados por el equipo 1.
13. team1-local-average: Media de partidos ganados como local entre los jugados como local por el equipo1.
14. team1-average-ppm: media de puntos por partidos obtenidos por el equipo 1

15. team1-streak: Racha de victorias seguidas por el equipo 1.
16. team1-players-value: Suma de la valoración de los jugadores del equipo 1.
17. team1-average-player-value: Valoración media de los jugadores del equipo 1.
18. percent-team2-seasonWins: Porcentaje de partidos que ha ganado el equipo2 durante la temporada.
19. percent-team2-seasonVisitorWins: Porcentaje de partidos que ha ganado el equipo2 durante la temporada jugando como visitante.
20. percent-team2-seasonVisitorDraws: Porcentaje de partidos que ha empatado el equipo2 durante la temporada jugando como visitante.
21. percent-team2-seasonVisiorDefeats: Porcentaje de partidos que ha perdido el equipo2 durante la temporada jugando como visitante.
22. team2-average: Media de partidos ganados frente a jugados por el equipo 2.
23. team2-local-average: Media de partidos ganados como local entre los jugados como local por el equipo2.
24. team2-average-ppm: media de puntos por partidos obtenidos por el equipo 2
25. team2-streak: Racha de victorias seguidas por el equipo 2.
26. team2-players-value: Suma de la valoración de los jugadores del equipo 2.
27. team2-average-player-value: Valoración media de los jugadores del equipo 2.

Para que WEKA pueda reconocer y tratar adecuadamente los atributos de cada instancia, éstos están en formato numérico.

Una vez determinados los atributos que utilizaremos para generar nuestro modelo, vamos a realizar un análisis de los mismos para cuantificar la importancia o relevancia de estos parámetros. Para esta tarea vamos a utilizar la interfaz gráfica de WEKA, concretamente la pestaña *select attributes*, la cual está dedicada al análisis de los atributos de las instancias. Dentro de esta pestaña podemos seleccionar dos tipos de algoritmos para evaluar los atributos, el primer algoritmo y más importante es el *attribute evaluator*, situado en la parte superior izquierda de la interfaz, esta opción nos permite seleccionar el algoritmo de evaluación de los atributos, condicionando la selección del segundo algoritmo, ya que no todos los algoritmos disponibles en el apartado *search method* se pueden utilizar con todos los algoritmos de la opción anterior. La funcionalidad de la opción *search method*" es básicamente para organizar la muestra de información del algoritmo utilizado en la opción *attribute evaluator*.

En la figura 23 se muestra la situación de estas opciones en la interfaz de la herramienta.

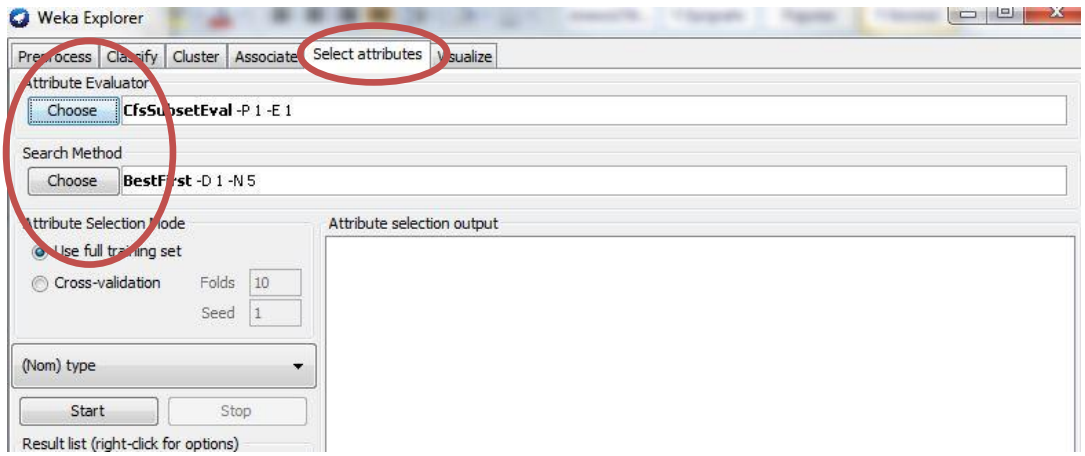

Figura 23: Weka select attributes

Entre los algoritmos disponibles para evaluar los atributos, vamos a utilizar dos de ellos, "InfoGainAttributeEval" porque este algoritmo evalúa la relevancia de un atributo con respecto a la clase, que en nuestro caso será el atributo encargado de clasificar las instancias; el otro atributo que vamos a utilizar es el "CfsSubSetEval", el cual evalúa el valor de un subconjunto de atributos considerando la capacidad predictiva individual de cada parámetros, junto con el grado de redundancia entre ellos.

Para evaluar los distintos atributos hemos generado un fichero de instancias con todos los partidos almacenados de la Liga BBVA y de la Liga Adelante, obteniendo los siguientes resultados:

Algoritmo InfoGainAttributeEval + Ranker

La combinación de estos algoritmos nos devolverá la relevancia de cada atributo ordenado de mayor a menor.

InfoGain	Attribute	Name
0.19176	9	percent-team1-seasonLocalWins
0.18842	21	percent-team2-seasonVisiorDefeats
0.17216	19	percent-team2-seasonVisitorWins
0.17083	11	percent-team1-seasonLocalDefeats
0.16129	10	percent-team1-seasonLocalDraws
0.15868	20	percent-team2-seasonVisitorDraws
0.14607	13	team1-local-average
0.13829	15	team1-streak
0.13339	8	percent-team1-seasonWins
0.12831	25	team2-streak
0.11508	24	team2-average-ppm
0.10261	18	percent-team2-seasonWins
0.07226	14	team1-average-ppm
0.07045	12	team1-average
0.06417	22	team2-average
0.01204	26	team2-players-value
0.012	1	team1-id
0.00906	2	team2-id
0.0088	23	team2-local-average
0.00874	16	team1-players-value
0.0079	17	team1-average-player-value
0.00458	27	team2-average-player-value
0	6	percent-team1-win
0	7	percent-team2-win
0	3	percent-match-local-win
0	4	percent-match-local-defeat
0	5	percent-match-draw

Tabla 24: InfoGain resultado

Como podemos apreciar en la tabla 24, de los 27 atributos que hemos definido cinco de ellos parecen no ejercer ningún tipo de influencia en la clasificación de la instancia, para asegurarnos y

evitar precipitarnos a la hora de deshacernos de estos atributos, vamos a comprobar qué nos devuelve el otro algoritmo selección.

Algoritmo CfsSubsetEval+BestFirst

Al combinar estos algoritmos obtendremos un listado con los atributos más relevantes de nuestro conjunto de entrenamiento.

CfsSubSetEval
percent-team1-seasonLocalWins
percent-team1-seasonLocalDraws
percent-team1-seasonLocalDefeats
team1-average-ppm
team1-streak
percent-team2-seasonWins
percent-team2-seasonVisitorWins
percent-team2-seasonVisitorDraws
percent-team2-seasonVisiorDefeats
team2-streak

Tabla 25: CfsSubsetEval resultado

Como podemos ver en la tabla 25, este algoritmo nos devuelve los atributos que ha considerado más importantes, en este caso son 10 atributos, que curiosamente coinciden con los 10 primeros atributos del algoritmo anterior, lo que parece confirmar los resultados de la evaluación anterior.

4.1.3. Salidas

Para poder clasificar correctamente las instancias y teniendo en cuenta el trabajo anterior a este proyecto, hemos decidido reservar una atributo de clase que utilizará la siguiente codificación:

Victoria local	1
Empate	X
Victoria visitante	2

Tabla 25: Codificación instancias

El motivo de utilizar solamente un atributo de clase para clasificar los posibles resultados de un partido, es debido a WEKA, ya que esta herramienta funciona más cómodamente con un solo atributo de clase en sus clasificadores.

4.2.Modelos entrenados

Como hemos explicado anteriormente debido a la naturaleza de las redes neuronales y del algoritmo de *backpropagation*, nos hemos visto obligados a desarrollar un plan de pruebas basado en el conocido método de prueba y error.

Para establecer el plan de pruebas hemos establecido cinco atributos que se modificarán según sea necesaria para cubrir todas la combinaciones que hemos creído oportunas. Estas variables son:

Atributo	Motivaciones
Épocas	Como no existe un número de épocas que podamos establecer a priori, hemos decidido utilizar como mínimo 4 rangos de épocas, aunque en algunos casos hemos creído oportuno aumentar el rango, los valores utilizados son 100, 300, 500 y 1000.
Temporadas de entrenamiento	Hemos querido comprobar el impacto en los resultados al utilizar dos conjuntos de entrenamiento con un número diferente de temporadas.
Filtrado de atributos	Puesto que el análisis de los parámetros de las instancias nos ha revelado que existen parámetros con influencia nula en la clasificación de la instancia, hemos querido comprobarlo empíricamente.
MLP común para las dos divisiones	Queremos comprobar si el uso de un modelo para cada una de las divisiones del fútbol español resulta ser un factor decisivo.
Tasa de aprendizaje	Hemos usado cinco valores distintos para la tasa de aprendizaje, 0'1, 0'3, 0'5, 0'7 y 0'9.

Tabla 26: Descripción variables plan de pruebas

Con estas variables esperamos desarrollar un plan de pruebas los suficientemente completo como para dar con un predictor lo suficientemente bueno como para obtener beneficios económicos con sus predicciones.

Para llevar a cabo la batería de pruebas hemos utilizado nuestra herramienta, con una ligeras modificaciones que en principio no estarán disponibles en la versión final debido a lo tedioso de su configuración. La modificación de la herramienta consiste en automatizar la generación de los modelos con todas las combinaciones posibles de las opciones anteriores, aunque aún así ha sido necesaria la intervención humana en algunos casos para poder generar los modelos.

Se han generado los siguientes ficheros de entrenamiento y test:

Nombre Fichero	Descripción
Train_Common.arff	Fichero de entrenamiento con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 1992-93 hasta la 2009-10.
Test_Common.arff	Fichero de test con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 2010-11 hasta la 2012-13.
Train_Common_AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 1992-93 hasta la 2009-10, con filtrado de atributos.
Test_Common_AttFiltered.arff	Fichero de test con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 2010-11 hasta la 2012-13, con filtrado de atributos.
Train_Common_B.arff	Fichero de entrenamiento con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 1992-93 hasta la 2007-08.
Test_Common_B.arff	Fichero de test con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 2008-09 hasta la 2012-13.
Train_Common_B_AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga

	BBVA y de la Liga Adelante, desde la temporada 1992-93 hasta la 2007-08, usando algoritmo de filtrado de atributos.
Test_Common_B_AttFiltered.arff	Fichero de test con partidos de la Liga BBVA y de la Liga Adelante, desde la temporada 2008-09 hasta la 2012-13, usando algoritmo de filtrado de atributos.
Train_1div.arff	Fichero de entrenamiento con partidos de la Liga BBVA, desde la temporada 1992-93 hasta la 2009-10.
Test_1div.arff	Fichero de test con partidos de la Liga BBVA, desde la temporada 2010-11 hasta la 2012-13.
Train_1div_AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga, desde la temporada 1992-93 hasta la 2009-10, con filtrado de atributos.
Test_1div_AttFiltered.arff	Fichero de test con partidos de la Liga BBVA, desde la temporada 2010-11 hasta la 2012-13, con filtrado de atributos.
Train_2div.arff	Fichero de entrenamiento con partidos de la Liga Adelante, desde la temporada 1992-93 hasta la 2009-10.
Test_2div.arff	Fichero de test con partidos de la Liga Adelante, desde la temporada 2010-11 hasta la 2012-13.
Train_2div_AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga Adelante, desde la temporada 1992-93 hasta la 2009-10, con filtrado de atributos.
Test_2div_AttFiltered.arff	Fichero de test con partidos de la Liga Adelante, desde la temporada 2010-11 hasta la 2012-13, con filtrado de atributos.
Train_1_B_div.arff	Fichero de entrenamiento con partidos de la Liga BBVA, desde la temporada 1992-93 hasta la 2007-08
Test_1div_B.arff	Fichero de test con partidos de la Liga BBVA, desde la temporada 2008-09hasta la 2012-13.
Train_1div_B _AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga, desde la temporada 1992-93 hasta la 2008-09, con filtrado de atributos.
Test_1div_B _AttFiltered.arff	Fichero de test con partidos de la Liga BBVA, desde la temporada 2007-08 hasta la 2012-13, con filtrado de atributos.
Train_2div_B.arff	Fichero de entrenamiento con partidos de la Liga Adelante, desde la temporada 1992-93 hasta la 2007-08
Test_2div_B.arff	Fichero de test con partidos de la Liga Adelante, desde la temporada 2008-09hasta la 2012-13.
Train_2div_B _AttFiltered.arff	Fichero de entrenamiento con partidos de la Liga Adelante, desde la temporada 1992-93 hasta la 2007-08, con filtrado de atributos.
Test_2div_B _AttFiltered.arff	Fichero de test con partidos de la Liga Adelante, desde la temporada 2008-09 hasta la 2012-13, con filtrado de atributos.

Tabla 27: Ficheros generados

Como podemos comprobar la cantidad de modelos generados es bastante grande, así que para poder llevar un control organizado de los modelos hemos utilizado una tabla que nos ayudará en la tarea de identificar qué modelos merecen la pena analizar, a continuación mostramos esta tabla con los mejores modelos que hemos encontrado, la tabla resumen completa se encuentra en el Anexo I: Tabla resumen modelos generados:

Observando la tabla 28, podemos comprobar que todos los modelos seleccionados poseen una tasa de aprendizaje de 0'7, no han utilizado filtrado de atributos y han sido generados con el mismo fichero de entrenamiento.

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,76	27,23
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	71,1	28,89
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,23	27,76
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	67,96	32,03
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	71,84	28,15
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,58	28,15
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,84	29,15
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,84	29,15
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,25	28,74
0,7	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,72	29,27

Tabla 28: Selección de modelos

5. Evaluación

Los resultados obtenidos por los modelos de predicción entrenados en el apartado anterior, se refieren a la probabilidad de acierto del resultado de un partido, sin embargo esto no es exactamente lo que estamos buscando, aunque estos resultados nos ayuda a estimar qué modelos nos pueden servir mejor para lo que realmente buscamos, un predictor de Quinielas.

Para poder evaluar apropiadamente nuestros modelos, hemos seleccionado varias jornadas de La Quiniela de la temporada 2012-13. Observando el reparto de los premios de las distintas jornadas que componen la temporada 2012-13, las hemos clasificado en dos tipos de Quiniela: Quiniela atípica y Quiniela regular. Las Quinielas atípicas son aquellas Quinielas en las que no ha habido acertantes de la categoría Pleno al 15, ya sea por la dificultad de la Quiniela o por las sorpresas producidas a lo largo de la jornada, este tipo de jornadas no suele ser muy habitual, las jornadas seleccionadas de dentro de esta categoría son las pertenecientes a las jornadas 2, 10, 17 y 37. Por otro lado las Quinielas catalogadas como regulares son aquellas Jornadas en las que hay acertantes de Pleno al 15 (usaremos la 18, 24 y 46).

La modalidad de la Quiniela utilizada para evaluar los modelos seleccionados ha sido la Apuesta Simple, sin recurrir a combinaciones de resultados múltiples, ciñéndonos a la apuesta mínima, que consiste en dos columnas simples de 0'5€ cada una, por lo que el precio del boleto asciende a 1€.

5.1.Pruebas de evaluación

Para poder visualizar de forma adecuada los resultados hemos diseñado la siguiente tabla, en la cual podemos ver el número de aciertos y premios obtenidos con respecto a una de las columnas apostadas. Ambas columnas disponen de la misma combinación, puesto que el modelo siempre genera la misma salida para la mista entrada.

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	66,66	6	33,33	3					Quiniela 10		
0,7	100	1992-93/2009-10					50	3	50	3	Quiniela 10	9	0
0,7	300	1992-93/2009-10	88,88	8	11,11	1					Quiniela 10		
0,7	300	1992-93/2009-10					50	3	50	3	Quiniela 10	11	140,24
0,7	500	1992-93/2009-10	88,88	8	11,11	1					Quiniela 10		
0,7	500	1992-93/2009-10					50	3	50	3	Quiniela 10	11	140,24
0,7	100	1992-93/2009-10	55,55	5	44,44	4	33,33	2	66,66	4	Quiniela 10	7	0
0,7	300	1992-93/2009-10	44,44	4	55,55	5	16,66	1	83,33	5	Quiniela 10	5	0
0,7	500	1992-93/2009-10	55,55	5	44,44	4	33,33	2	66,66	4	Quiniela 10	7	0
0,7	1000	1992-93/2009-10	44,44	4	55,55	5	33,33	2	66,66	4	Quiniela 10	6	0

Tabla 29: Resumen evaluaciones Jornada 10

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	77,77	7	22,22	2					Quiniela 17		
0,7	100	1992-93/2009-10					0,5	3	0,5	3	Quiniela 17	10	7,8
0,7	300	1992-93/2009-10	88,88	8	11,11	1					Quiniela 17		
0,7	300	1992-93/2009-10					66,66	4	33,33	2	Quiniela 17	12	121,93
0,7	500	1992-93/2009-10	88,88	8	11,11	1					Quiniela 17		
0,7	500	1992-93/2009-10					66,66	4	33,33	2	Quiniela 17	12	121,93
0,7	100	1992-93/2009-10	66,66	6	33,33	3	66,66	4	33,33	2	Quiniela 17	10	7,8
0,7	300	1992-93/2009-10	66,66	6	33,33	3	66,66	4	33,33	2	Quiniela 17	10	7,8
0,7	500	1992-93/2009-10	77,77	7	22,22	2	66,66	4	33,33	2	Quiniela 17	11	12,98
0,7	1000	1992-93/2009-10	77,77	7	22,22	2	66,66	4	33,33	2	Quiniela 17	11	12,98

Tabla 30: Resumen evaluación Jornada 17

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	55,55	5	44,44	4					Quiniela 18		
0,7	100	1992-93/2009-10					33,33	2	66,66	4	Quiniela 18	7	0
0,7	300	1992-93/2009-10	66,66	6	33,33	3					Quiniela 18		
0,7	300	1992-93/2009-10					50	3	50	3	Quiniela 18	9	0
0,7	500	1992-93/2009-10	66,66	6	33,33	3					Quiniela 18		
0,7	500	1992-93/2009-10					50	3	50	3	Quiniela 18	9	0
0,7	100	1992-93/2009-10	55,55	5	44,44	4	66,66	4	33,33	2	Quiniela 18	9	0
0,7	300	1992-93/2009-10	66,66	6	33,33	3	50	3	50	3	Quiniela 18	9	0
0,7	500	1992-93/2009-10	66,66	6	33,33	3	83,33	5	16,66	1	Quiniela 18	11	2,36
0,7	1000	1992-93/2009-10	66,66	6	33,33	3	83,33	5	16,66	1	Quiniela 18	11	2,36

Tabla 31: Resumen evaluaciones Jornada 18

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	100	8	0	0					Quiniela 2		
0,7	100	1992-93/2009-10					100	7	0	0	Quiniela 2	15	1318419,08
0,7	300	1992-93/2009-10	75	6	25	2					Quiniela 2		
0,7	300	1992-93/2009-10					100	7	0	0	Quiniela 2	13	159808,36
0,7	500	1992-93/2009-10	75	6	25	2					Quiniela 2		
0,7	500	1992-93/2009-10					100	7	0	0	Quiniela 2	13	6308,22
0,7	100	1992-93/2009-10	75	6	25	2	71,42	5	28,5	2	Quiniela 2	11	518,3
0,7	300	1992-93/2009-10	100	8	0	0	100	7	0	0	Quiniela 2	15	1318419,08
0,7	500	1992-93/2009-10	100	8	0	0	100	7	0	0	Quiniela 2	15	1318419,08
0,7	1000	1992-93/2009-10	87,5	7	12,5	1	71,42	5	28,5	2	Quiniela 2	12	6308,22

Tabla 32 Resumen evaluaciones Jornada 2

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	50	4	50	4					quiniela 24		
0,7	100	1992-93/2009-10					42,85	3	57,14	4	quiniela 24	7	0
0,7	300	1992-93/2009-10	62,5	5	37,5	3					quiniela 24		
0,7	300	1992-93/2009-10					57,14	4	42,85	3	quiniela 24	9	0
0,7	500	1992-93/2009-10	62,5	5	37,5	3					quiniela 24		
0,7	500	1992-93/2009-10					71,42	5	28,57	2	quiniela 24	10	0
0,7	100	1992-93/2009-10	50	4	50	4	71,42	5	28,5	2	quiniela 24	9	0
0,7	300	1992-93/2009-10	37,5	3	62,5	5	100	7	0	0	quiniela 24	10	0
0,7	500	1992-93/2009-10	37,5	3	62,5	5	100	7	0	0	quiniela 24	10	0
0,7	1000	1992-93/2009-10	50	4	50	4	71,42	5	28,5	2	quiniela 24	9	0

Tabla 33: Resumen evaluaciones jornada 24

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	44,44	4	55,55	5					quiniela 46		
0,7	100	1992-93/2009-10					66,66	4	33,33	2	quiniela 46	8	0
0,7	300	1992-93/2009-10	44,44	4	55,55	5					quiniela 46		
0,7	300	1992-93/2009-10					50	3	50	3	quiniela 46	7	0
0,7	500	1992-93/2009-10	33,33	3	66,66	6					quiniela 46		
0,7	500	1992-93/2009-10					66,66	4	33,33	2	quiniela 46	8	0
0,7	100	1992-93/2009-10	33,33	3	66,66	6	50	3	50	3	quiniela 46	6	0
0,7	300	1992-93/2009-10	33,33	3	66,66	6	33,33	2	66,66	4	quiniela 46	5	0
0,7	500	1992-93/2009-10	33,33	3	66,66	6	50	3	50	3	quiniela 46	6	0
0,7	1000	1992-93/2009-10	33,33	3	66,66	6	66,66	4	33,33	2	quiniela 46	7	0

Tabla 34: Resumen evaluaciones Jornada 46

Tasa Aprendizaje	Ciclos	Temp. Train	%acierto 1 División	Aciertos	%error 1 División	Aciertos	%acierto 2 División	Aciertos	%error 2 División	Aciertos	Jornada	Total Aciertos	Premio
0,7	100	1992-93/2009-10	12,5	1	87,5	7					Quiniela 37		
0,7	100	1992-93/2009-10					71,4	5	28,5	2	Quiniela 37	6	0
0,7	300	1992-93/2009-10	25	2	75	6					Quiniela 37		
0,7	300	1992-93/2009-10					71,4	5	28,5	2	Quiniela 37	7	0
0,7	500	1992-93/2009-10	25	2	75	6					Quiniela 37		
0,7	500	1992-93/2009-10					71,4	5	28,5	2	Quiniela 37	7	0
0,7	100	1992-93/2009-10	25	2	75	6	85,71	6	14,28	1	Quiniela 37	8	0
0,7	300	1992-93/2009-10	25	2	75	6	85,71	6	14,28	1	Quiniela 37	8	0
0,7	500	1992-93/2009-10	25	2	75	6	71,4	5	28,5	2	Quiniela 37	7	0
0,7	1000	1992-93/2009-10	37,5	3	62,5	5	57,14	4	42,85	3	Quiniela 37	7	0

Tabla 35: Resumen evaluaciones jornada 37

La siguiente tabla (tabla 36) muestra de forma resumida los resultados de la evaluación, llama la atención el volumen de premios obtenidos en las jornadas utilizadas para la evaluación de los modelos. Observando la tabla llegamos a la conclusión de que utilizar un modelo de predicción independiente para cada división, obtiene mejores resultados, sin embargo los modelos utilizados con peores medias han obtenido los premios más elevados, por lo que es necesario seguir trabajando con los datos para intentar obtener modelos más fiables.

Sin embargo, nos quedamos con la estupenda media de aciertos que ha cosechado el mejor de los modelos, ya que una media de 10 aciertos en 15 partidos, con apuestas simples es admirable, un hito complicado incluso para expertos en la materia.

Tasa Aprendizaje	Ciclos	Temp. Train	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	Premio	Promedio
0,7	100	1992-93/2009-10	false	a	false 1ª Div	28		
0,7	100	1992-93/2009-10	false	a	false 2ª Div	28	1318426,88	8,857142857
0,7	300	1992-93/2009-10	false	a	false 1ª Div	28		
0,7	300	1992-93/2009-10	false	a	false 2ª Div	28	160070,53	9,714285714
0,7	500	1992-93/2009-10	false	a	false 1ª Div	28		
0,7	500	1992-93/2009-10	false	a	false 2ª Div	28	6570,39	10
0,7	100	1992-93/2009-10	false	a	true	28	526,1	8,571428571
0,7	300	1992-93/2009-10	false	a	true	28	1318426,88	8,857142857
0,7	500	1992-93/2009-10	false	a	true	28	1318434,42	9,571428571
0,7	1000	1992-93/2009-10	false	a	true	28	6323,56	9

Tabla 36: Tabla resumen evaluación

6. Conclusiones y trabajos futuros

6.1. Conclusiones

Observando los resultados obtenidos en el apartado anterior, vemos que todos los modelos han resultado rentables, destacando aquellos que obtuvieron al menos un pleno al 15 y que han proporcionado un rendimiento económico mucho más elevado del esperado inicialmente.

La siguiente tabla muestra la Jornada 2 utilizada en la evaluación de los modelos:

Local	Visitante	1	x	2
Betis	R.Vallecano			2
Espanyol	Zaragoza			2
Málaga	Mallorca		X	
R.Sociedad	Celta	1		
Osasuna	Barcelona			2
Getafe	Real Madrid	1		
Granada	Sevilla		X	
Sabadell	Villarreal		X	
Guadalajara	Girona			2
Ponferradina	Alcorcón	1		
Hércules	Elche			2
Córdoba	R.Santander	1		
Sp.Gijón	Murcia			2
Huesca	Numancia		X	
Valencia	Dep.Coruña		X	

Tabla 37: Jornada 2 Quiniela 2012-13

Analizando la tabla 37, observamos que esta Quiniela en concreto tiene más partidos de La Liga Adelante de lo habitual, factor que dificulta la labor de predicción para un usuario medio, ya que la Liga Adelante no se sigue con tanta frecuencia como La Liga BBVA y además parece que se han producido resultados a priori inesperados, como la derrota del Real Madrid.

Por estos motivos entendemos que no haya habido acertantes ni de Pleno al 15 ni de 1ª Categoría. Sin embargo varios de los modelos de predicción parecen haber funcionado correctamente en este caso, como lo han hecho la mayoría de los modelos al realizar predicciones más acertadas en las jornadas atípicas que en la jornadas regulares.

El hecho de que los predictores realicen predicciones más acertadas con quinielas atípicas, nos hace dudar sobre la fiabilidad de los modelos, si bien es cierto que como esperábamos, prácticamente con todos los modelos entrenados hemos obtenido un determinado rendimiento económico, también es cierto que esta no era esta la forma en la que se esperaba que se comportaran los modelos, ya que a priori se suponía que los modelos

funcionarían mejor con quinielas en las que no hubiese sorpresas y por lo tanto hubiese un mayor número de acertantes, obteniendo premios más moderados.

Por este motivo es posible que sea necesaria una nueva revisión de los datos utilizados para elaborar los modelos de predicción, elaborar nuevas hipótesis y probablemente encontrar nuevos parámetros que puedan ser de utilidad.

6.2.Trabajos futuros

Las secciones anteriores estaban orientadas a analizar el trabajo realizado en este proyecto . A continuación se proponen futuras líneas de trabajo e investigación.

- La primera opción a tener en cuenta es la de completar información de la base de conocimiento, como podría ser la inclusión de los árbitros, estadísticas más completas de los jugadores, factores meteorológicos o económicos (presupuesto de los equipos, etc).
- Otra alternativa a tener en cuenta podría ser el estudio de series temporales, ya que las temporadas a simple vista presentan patrones temporales comunes, vacaciones invierno, final de temporada, competiciones ajenas, etc. que pueden afectar a los resultados de los encuentros.
- También sería recomendable generar modelos capaces de generar salidas complejas, permitiendo combinaciones de resultados en la salida, obteniendo predicciones de dos o tres posibles resultados para partidos muy igualados o inciertos.

7. Anexo I: Tabla resumen modelos generados

Para determinar el número de capas ocultas de cada modelo, se ha utilizado una de las opciones que ofrece WEKA por defecto, y que nos ha parecido oportuna utilizar. El número de capas ocultas utilizadas equivale a la mitad, redondeando hacia arriba, del número de atributos que componen cada una de las instancias utilizadas en la fase de entrenamiento.

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,1	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	74,605 (567)	25,495 (193)
0,1	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	71,42(660)	28,57(264)
0,1	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	73,815(561)	26,184(199)
0,1	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	70.238(649)	29.76(275)
0,1	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	74,078(563)	25,921(197)
0,1	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	70,02(647)	29,97(277)
0,1	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	73,421(558)	26,578(202)
0,1	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	70.88(655)	29.11(269)

Tabla 38: Tabla resumen modelos entrenados, parte 1

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,3	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	74,47 (566)	25,52 (194)
0,3	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	71,75(663)	28,24(261)
0,3	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	73,94(562)	26,95(198)
0,3	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,91(646)	30,08(278)
0,3	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	73,684(560)	26,31(200)
0,3	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,58(643)	30,41(281)
0,3	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	73,15(556)	26,84(204)
0,3	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,37(6419	30,62(283)

Table 39: Tabla resumen modelos entrenados, parte 2

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,5	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	74,07(563)	25,92(197)
0,5	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	70,12(648)	29,87(276)
0,5	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,89(554)	27,10(206)
0,5	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,37(641)	30,62(283)
0,5	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	71,84(546)	28,15(214)
0,5	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,48(642)	30,51(282)
0,5	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,89(554)	27,10(206)
0,5	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	68,72(635)	31,27(289)

Table 40: Tabla resumen modelos entrenados, parte 3

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,76(553)	27,23(207)
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	71,10(657)	28,89(267)
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,23(549)	27,76(211)
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	67,96(628)	32,03(296)
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	71,84(546)	28,15(214)
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,58(643)	28,15(281)
0,7	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	71,84(546)	28,15(214)
0,7	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	69,37(641)	30,62(283)

Tabla 41: Tabla resumen modelos entrenados, parte 4

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,9	100	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,36(550)	27,63(210)
0,9	100	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	68,29(631)	31,71(293)
0,9	300	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	71,57(544)	28,42(216)
0,9	300	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	68,39(632)	31,06(292)
0,9	500	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,10(548)	27,89(212)
0,9	500	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	67,31(622)	32,684(302)
0,9	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 1ª Div	28	72,10(548)	27,89(212)
0,9	1000	1992-93/2009-10	2010-11/2012-13	false	a	false 2ª Div	28	68,39(632)	31,60(292)

Tabla 42: Tabla resumen modelos entrenados, parte 5

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,1	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,80(1226)	27,19(458)
0,1	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,62(1223)	27,37(461)
0,1	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,44(1220)	27,55(464)
0,1	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,38(1219)	27,61(465)
0,3	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,14(1215)	27,85(469)
0,3	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	72,03(1213)	27,96(471)
0,3	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,73(1208)	28,26(476)
0,3	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,61(1206)	28,38(478)

Tabla 43: Tabla resumen modelos entrenados, parte 6

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,5	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,79(1209)	28,20(475)
0,5	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,43(1203)	28,56(481)
0,5	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,96(1195)	29,03(489)
0,5	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,90(1194)	29,09(490)
0,7	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,84(1193)	29,15(491)
0,7	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,84(1193)	29,15(491)
0,7	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	71,25(1200)	28,74(484)
0,7	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,72(1191)	29,27(493)

Tabla 44: Tabla resumen modelos entrenados, parte 7

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,9	100	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,07(1180)	29,92(504)
0,9	300	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,60(1189)	29,39(495)
0,9	500	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,48(1187)	29,51(497)
0,9	1000	1992-93/2009-10	2010-11/2012-13	false	a	true	28	70,72(1191)	29,27(493)

Tabla 45: Tabla resumen modelos entrenados, parte 8

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,1	100	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	72,92(1228)	27,07(456)
0,1	300	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,574(1239)	26,4252(445)
0,1	500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,93(1245)	26,06(439)
0,1	1000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,10(1248)	25,89(436)
0,1	1500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,52	25,41
0,1	2000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,7	25,26
0,1	2500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,94	25,05
0,1	3000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,88	25,11

Tabla 46: Tabla resumen modelos entrenado, parte 9

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,3	100	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	72,74	27,25
0,3	300	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	72,8	27,19
0,3	500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,04	26,95
0,3	1000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,15	26,84
0,3	1500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,27	26,72
0,3	2000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,21	26,78

Tabla 47: Tabla resumen modelos entrenado, parte 10

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,5	100	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,009	26,9
0,5	300	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,15	26,84
0,5	500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,574(1239)	26,4252(445)
0,5	1000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,63	26,35
0,5	1500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,574(1239)	26,48

Tabla 48: Tabla resumen modelos entrenados, parte 11

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,7	100	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,09	26,9
0,7	300	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,51	26,48
0,7	500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,45	26,54
0,7	1000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,87	26,12
0,7	1500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,10(1248)	25,89(436)
0,7	2000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	74,46	25,53
0,7	2500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,93(1245)	26,06(439)

Tabla 49: Tabla resumen modelos entrenados, parte 12

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,9	100	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,12	26,78
0,9	300	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,51	26,48
0,9	500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,51	26,48
0,9	1000	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,574(1239)	26,4252(445)
0,9	1500	1992-93/2009-10	2010-2011/2012-13	true cfsSubset+BestFirst	a	true	11	73,57	26,4252(445)

Tabla 50: Tabla resumen modelos entrenados, parte 13

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,1	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,2	25,7
0,1	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,2	27,7
0,1	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,47	25,52
0,1	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,27	27,72
0,1	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,47	25,52
0,1	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,27	27,72
0,1	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,21	25,78
0,1	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,01	27,98

Tabla 51: Tabla resumen modelos entrenados, parte 14

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,3	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,64	26,31
0,3	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,44	28,51
0,3	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,15	26,84
0,3	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	70,95	29,04
0,3	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,55	26,44
0,3	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,35	28,64
0,3	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,68	26,31
0,3	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,48	28,51

Tabla 52: Tabla resumen modelos entrenados, parte 15

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,5	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,68	26,31
0,5	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,48	28,51
0,5	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	76,68	26,31
0,5	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	74,48	28,51
0,5	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,07	25,9
0,5	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,87	28,1
0,5	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,21	25,7
0,5	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,01	27,9

Tabla 53: Tabla resumen modelos entrenados, parte 16

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,7	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,28	26,7
0,7	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,08	28,9
0,7	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	72,89	27,1
0,7	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	70,69	29,3
0,7	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,68	26,31
0,7	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,48	28,51
0,7	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,81	26,18
0,7	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,61	28,38

Tabla 54: Tabla resumen modelos entrenados, parte 17

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,9	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,28	26,71
0,9	100	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,08	28,91
0,9	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,21	25,78
0,9	300	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,01	27,98
0,9	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	73,94	26,05
0,9	500	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	71,74	28,25
0,9	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 1ª Div	11	74,34	25,65
0,9	1000	1992-93/2009-10	2010-11/2012-13	true cfsSubset+BestFirst	a	false 2ª Div	11	72,14	27,85

Table 55: Tabla resumen modelos entrenados, parte 18

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,1	100	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,83(2453)	27,16(915)
0,1	300	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,83(2453)	27,16(915)
0,1	500	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,77(2451)	27,11(917)
0,1	1000	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,59(2445)	27.40(923)
0,3	100	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,26(2434)	27,73(934)
0,3	300	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,35(2437)	27,64(931)
0,3	500	1992-93/2007-08	2008-09/2012-13	false	a	true	28	72,09%(2428)	27,9%(940)
0,3	1000	1992-93/2007-08	2008-09/2012-13	false	a	true	28	71,79(2418)	28,20(950)

Table 56: Tabla resumen modelos entrenados, parte 19

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,5	100	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	71,52(2409)	28,47(959)
0,5	300	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	71,55(2410)	28,44(958)
0,5	500	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	71,58(2411)	28,41(957)
0,5	1000	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	71,64(2413)	28,35(955)
0,7	100	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	70,04(2359)	29,95(1009)
0,7	300	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	70,10(2361)	29,89(1007)
0,7	500	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	69,89(2354)	30,10(1014)
0,7	1000	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	69,98(2357)	30,01(1011)

Tabla 57: Tabla resumen modelos entrenados, parte 20

Tasa Aprendizaje	Ciclos	Temp. Train	Temp Test	AttributeSelection	Número de capas ocultas	common NN	Attributos de entrada	%acierto	%error
0,9	100	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	69,26(2333)	30,73(1035)
0,9	300	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	69,80(2351)	30,19(1017)
0,9	500	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	70,014(2359)	29,95(1009)
0,9	1000	1992-93/2007-08	2008-09/2012-13	true cfsSubset+BestFirst	a	true	10	70,16(2363)	29,83(1005)

Tabla 58: Tabla resumen modelos entrenados, parte 21

8. Bibliografía

[1] 2014. **BDFutbol** Es Una Web Independiente Que Tiene Como Objetivo La Recopilación Exhaustiva De Todos Los Datos y Estadísticas Relativos Al Fútbol, Mostrándolos De Forma Ordenada, Estructurada y Accesible. Available from: http://www.bdfutbol.com/es/index.html.

[2] 2014. Página Oficial De Loterías y Apuestas Del Estado. Available from: http://loteriasyapuestas.es/.

[3] Available from: http://jsoup.org/.

[4] BOEHM, B., 1986. A Sprial Model of Software Development and Enhancement. .

[5] COCKBURN, A. Agile Software Development. Highsmith Series.

[6] DIETTERICH, T.G., 2000. Ensemble Methods in Machine Learning. Springer-Verlag London, UK.: Berlin.

[7] Fayyad, U. M Piatetsky-Shapiro, G., & Smyth, P., 1996. From Data Mining to Knowledge Discovery: An Overview.

[8] ganjisaffar@gmail.com. Crawler4j is an Open Source Project; Please use our Forum to Ask any Question. You can also Open an Issue for any Bug Or Feature You Want to Request. but First Read the Examples Below as Well as the Wiki to Better Understand Your Way Around Your Crawler. Available from: https://code.google.com/p/crawler4j/.

[9] GUILLOT, L., 2014. Blog Personal De Laura Guillot. Available from: http://www.lauraguillot.com/2008/06/victor-el-predictor-predice-que-espaa.html.

[10] MANNING, C.D., RAGHAVAN, P. and SCH¸TZE, H., 2008. Introduction to Information Retrieval. Cambridge University Press. In: . Introduction to Information Retrieval. Cambridge University Press, pp. Capitulo 20.

[11] olbapordep@terra.com., 2014. Available from: http://liga.host56.com/.

[12] SÁNCHEZ FUENTES, M.Á., FUENTES LÓPEZ, S. and FUENTES LÓPEZ, J.A., 2012. Mysql. 1ª ed. Don Benito: Editorial Edita ISBN 978-84-9997-879-6.

[13] The Waikato University., 2014. WEKA Offcial Site. Available from: http://www.cs.waikato.ac.nz/ml/index.html.

[14]W3C., 2014. Available from: http://www.w3.org/2005/03/DOM3Core-es/introduccion.html.

[15] SORIA, E. and BLANCO, A. Redes Neuronales Artificiales.

[16] Available from: http://en.wikipedia.org/wiki/Kelly_criterion.

[17] Available from: http://en.wikipedia.org/wiki/Martingale_%28probability_theory%29.

[18] VEGAS GARCÍA, A., SUÁREZ CETRULO, A.L.and PÉREZ SÁNCHEZ, F.J. Programmed Software Based on Neural Networks (SNNS) used for Football Betting, 2012 (Java).

[19] Levenshtein Distance. Available from: http://en.wikipedia.org/wiki/Levenshtein_distance.